UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

# NoMoDEI

A framework for
**No**rm **Mo**nitoring
on
**D**ynamic **E**lectronic **I**nstitutions

## P.H.D Thesis

PhD Candidate: Ignasi Gómez Sebastià

Advisor: Javier Vázquez Salceda

Artificial Intelligence PhD Program
KEMLG: Knowledge Engineering and Machine Learning Group November 2015

# Contents

# List of Figures

# Acknowledgements

I would like to start thanking my work-mates (both actual and former) Luis, Sergio, Dario, Arturo, Ana, Bea, Jonathan, Cristian, Juan Carlos, Guillem, Roberto, Miquel, Sofia, Victor and everyone I had the honour to collaborate with in several projects. They managed to make me feel like at home when being at our workplace. Special thanks to Luis, Bea and Sergio who shared with me (both in time and space) the interesting process of writing a Thesis document.

Thanks to the KEMLG group in general and Javier in particular for giving me the chance to collaborate with them when I was still a Master student, and allowing me to discover how does it feel like to be a researcher. Special thanks to Javier, Ulises, Arturo, Luis, Sergio and Dario for the unforgettable experience of collaborating with them in several projects. Thanks to them I am now a better professional.

Also thanks to Dr. Juan Carlos Nieves, who patiently corrected me several mistakes when writing my very first papers.

I would like to thank as well Prof. Manel Poch, Dr. Marta Verdaguer and Mr. Luis Oliva for their help defining the wastewater management use case.

I would also like to thank Dr. Cristian Barrué for his help defining the Health Care use case, and for allowing me to test NoMoDEI in the *AVICENA* scenario.

I would like to thank Sergio Alvarez-Napagao, Dario Garcia and Jonathan Moreno, for allowing me to participate in the development of COAALAS which was a great source of interesting work and ideas.

Also thanks to the reviewers for the time they spent in reading and assessing the thesis. I may not know them, but their feedback has been especially useful for the improvement of the document.

I would like to thank too Prof. Ulises Cortés, for his useful opinions on this document and his help on defining both use cases. And special thanks for his encouragement while writing this document, his sense of humour has been a source of laughs on some tough days. Also thanks to him for allowing me to discover the joy of lecturing.

I would like to thank my advisor Javier Vázquez-Salceda for his valuable help and guidance in developing and reviewing this document. This document would have been much harder to write without such a brilliant advisor. Also thanks for his inestimable help with several bureaucratic procedures which I seem to be allergic to.

I would also like to thank Sergio Alvarez-Napagao for his patient help explaining me the original monitoring framework, both regarding the formal concepts and the implementation. Also for his valuable opinions on where to extend it, and finally for his pristine clear code which allowed for an easy and efficient expansion of the original software. I would like to say his development methodology is a mirror, not only for being crystal clear, but also for being something I would like to reflect on my own code. Furthermore, working hand to hand with him has been a real honour.

I would like to acknowledge the following projects which I have participated in during the development of this Thesis. They have partially funded this Thesis and allowed me to make a living of my passion:

- **ALIVE**: Coordination, Organisation and Model Driven Approaches for Dynamic, Flexible, Robust Software and Services Engineering. Founded by the European Comission (ICT-FP7-215890)
- **RAISME**: Application Innovation for SMEs. Founded by the European Comission (FP7-SME-2010-1-262469)
- **SUPERHUB**: SUstainable and PERsuasive Human Users moBility in future cities. Founded by the European Comission (ICT-FP7- 289067)

And finally and specially, to my family, including but not limited to my parents Dori and Francisco. They instilled in me the love for learning, reading and culture. And of course, this work would not be possible without them.

IGNASI GÓMEZ-SEBASTIÀ
Barcelona
November 2015

# Abstract

With the growth of the Internet, computational systems have become more and more complex, often including complicate interconnected networks of autonomous components. The need to bring some organisational structure into autonomous systems becomes urgent, as this allows regulating the behaviour of the different autonomous components to ensure their objectives are aligned with the holistic objectives of the system.

Normative Systems are one of the mechanisms that can be applied to define and enforce acceptable behaviour within distributed electronic systems which should comply with some (human) regulations. One of the requirements to effectively implement Normative Systems is to be able to assess, at runtime, the state of the normative environment. Existing lines of research have already tried to tackle this issue on some simple scenarios. However, more complex scenarios may appear, for instance, scenarios where the normative context is not static, but it expands and contracts as new norms are added to the institution and removed from it respectively.

As in human legal systems, it is easy to foresee that some of these electronic normative environments will not be static. They should be able to evolve through time as regulations change, effectively adapting to new situations and behaviours. Under these conditions, a monitoring system must be able to continue computing the state of the normative environment at runtime, as often we can not afford to perform the changes on the normative context off-line. Furthermore, it must be guaranteed the monitoring system can keep producing states of the normative environment that are consistent with the changes performed on the normative context. For instance, if a norm has been removed from the normative context, it does not make sense anymore to compute normative states where the norm has been violated.

In this thesis we present **NoMoDEI**, a normative monitoring framework for dynamic Electronic Institutions. We formalize and develop an extended normative framework and architecture to cope with scenarios where the normative context is dynamic, therefore norms can be added, removed and updated. The operations are to be performed at runtime, without having to stop computing the normative state. The normative states computed are consistent with the expansion and contraction operations.

**NoMoDEI** is introduced in three steps. First, we formally define the operations to be supported in order to allow for expanding and contracting the normative context. Then, we instantiate the formal operations, providing implementation details. Finally, we demonstrate our framework by applying it to two use cases: E-health systems and waste-water management on a river basin.

# Resumen

Con el auge de Internet, los sistemas computacionales se han vuelto mucho más complejos, a menudo incorporando complicadas redes interconectadas de componentes autónomos. Es por ello que la necesidad de incorporar estructuras organizacionales en los sistemas autónomos se acentúa, dado que esto permite regular el comportamiento de los diferentes componentes autónomos para asegurar que sus objetivos se encuentran alineados con los objetivos generales del sistema.

Los Sistemas Normativos (*i.e.* Normative Systems) son uno de los mecanismos que podemos aplicar para definir e imponer patrones aceptables de comportamiento en sistemas electrónicos distribuidos. Esto es especialmente importante cuando el sistema es regido por regulaciones (normalmente humanas). Uno de los requisitos para implementar Sistemas Normativos es ser capaz de determinar, en tiempo de ejecución, el estado del entorno normativo. Existen líneas de investigación que ya han abordado este problema en algunos escenarios simples. Sin embargo, el mundo real nos provee de escenarios más complejos, por ejemplo, escenarios donde el contexto normativo no es estático, si no que se expande y contrae a medida que nuevas normas son añadidas o eliminadas de la institución.

Justo como ocurre con los sistemas legales humanos, es fácil prever que algunos de estos contextos normativos electrónicos no serán estáticos. Estos contextos deberían ser capaces de evolucionar a través del tiempo a medida que las regulaciones cambian, adaptándose a nuevas situaciones y comportamientos. Bajo estas condiciones, un sistema de monitorización debe ser capaz de continuar calculando el estado del entorno normativo en tiempo de ejecución. Esto viene dado porque a menudo no vamos a poder permitirnos realizar los cambios en el entorno normativo deteniendo el proceso de monitorización. Es más, se debe garantizar que el sistema de monitorización puede ser capaz de seguir produciendo estados del entorno normativo consistentes con los cambios realizados. Por ejemplo, el hecho de eliminar una norma hace que no tenga mucho sentido continuar calculando estados normativos donde esa norma ha sido violada.

En esta Tesis presentamos **NoMoDEI**, una infraestructura de monitorización normativa para instituciones electrónicas dinámicas. Formalizamos y desarrollamos una infraestructura de monitorización normativa extendida capaz de trabajar en escenarios donde el contexto normativo es dinámico. Es por ello que diversas normas pueden ser introducidas, eliminadas o actualizadas del contexto normativo en cualquier momento. Dichas operaciones deben ser realizadas en tiempo de ejecución, esto es, sin dejar de calcular el estado normativo. Es más, los estados normativos calculados deben ser consistentes con las respectivas operaciones de extensión o contracción del contexto.

Durante la Tesis, presentamos **NoMoDEI** en tres pasos. Para empezar, proveemos una definición formal de las operaciones que la infrastructura debe soportar para permitir expandir y contraer el contexto normativo. A continuación instanciamos dichas operaciones al proveer detalles de implementación. Finalmente, demostramos que nuestra infraestructura es capaz de ser aplicada a casos de uso reales introduciendo dos casos: sistemas de salud electrónicos (*i.e.* E-health) y sistemas de tratamiento de aguas residuales en la cuenca de un rio.

# Resum

Amb l'expansió d'Internet els sistemes computacionals han esdevingut més complexos, sovint incorporant complicades xarxes interconnectades de components autònoms. Es per això que la necessitat d'incorporar estructures organitzacionals en el sistemes autònoms s'accentua, donat que aquestes estructures permeten regular el comportament dels diferents components autònoms, tot assegurant que els seus objectius es troben alineats amb els objectius generals del sistema.

Els Sistemes Normatius (*i.e.* Normative Systems) són un dels mecanismes que podem aplicar per definir i imposar patrons acceptables de comportament dintre de sistemes electrònics distribuïts. Això esdevé especialment important quan el sistema es troba regimentat per regulacions (normalment humanes). Un dels requeriments per implementar Sistemes Normatius és ser capaços de determinar, en temps d'execució, l'estat de l'entorn normatiu. Existeixen línies de recerca que ja han tractat aquest problema en alguns escenaris simples. El món real però ens ofereix escenaris més complexes, com per exemple, escenaris on el context normatiu no és estàtic, si no que s'expandeix i contrau a mesura que noves normes són afegides o eliminades de la institució.

Tal com passa als sistemes legals humans, és fàcil preveure que alguns contextos normatius electrònics no seran estàtics. Aquests contextos haurien de ser capaços d'evolucionar a través del temps a mesura que les regulacions canvien, adaptant-se a noves situacions i comportaments. Sota aquestes condicions, un sistema de monitorització ha de ser capaç de continuar calculant l'estat de l'entorn normatiu en temps d'execució, ja que sovint no ens podem permetre realitzar els canvis a l'entorn normatiu aturant el procés de monitorització. És més s'ha de garantir que el sistema de monitorització sigui capaç de continuar produint estats de l'entorn normatiu de forma consistent amb els canvis realitzats. Per exemple, el fet d'eliminar una norma fa que no tingui gaire sentit continuar calculant estats normatius on aquesta norma ha estat violada.

A aquesta Tesi presentem **NoMoDEI**, una infraestructura de monitorització normativa per institucions electròniques dinàmiques. Formalitzem i desenvolupem una infraestructura de monitorització normativa estesa capaç d'operar en escenaris on el context normatiu es dinàmic. Es a dir, diverses normes poden ser introduïdes, eliminades o actualitzades del context normatiu en qualsevol moment. Aquestes operacions s'han de poder realitzar en temps d'execució, es a dir, sense deixar de calcular l'estat normatiu. Es més, els estats normatius calculats han de ser consistents amb les respectives operacions d'extensió o contracció del context.

Durant la Tesi presentem **NoMoDEI** en tres passos. Primer proporcionem una definició formal de les operacions que la infraestructura ha de suportar per permetre expandir i contraure el context normatiu. A continuació instanciem aquestes operacions proporcionant detalls d'implementació. Finalment demostrem que la nostra infraestructura pot ser aplicada a casos d'ús del món real introduint dos casos: sistemes de salut electrònics (*i.e.* E-health) i sistemes de tractament d'aigües residuals a la conca d'un riu.

# CHAPTER 1

# Introduction

With the growth of the Internet and the World Wide web, computational systems have become more and more complex. As systems grow to include more autonomous components (sometimes hundreds or thousands of them), often resulting in complicate interconnected networks of components, the need to bring some organisational structure into autonomous systems becomes urgent. One of the main motivations behind this is regulating the behaviour of the different autonomous components to ensure their objectives are aligned with the holistic objectives of the system. Therefore, implementing complex systems (*e.g.*, Multi-Agent Systems) able to cope with social and organisational guidelines becomes a necessity.

Several authors argue that the design of Multi-Agent Systems in complex (and specially open) environments can benefit from social abstractions in order to deal with problems in coordination, cooperation and trust among agents ( please notice that such problems are also present in human societies). These has been explored in various formalisations and there are many formal theoretical frameworks that allow to implement and operationalise social abstractions such as policies and norms. Such formalisations come in several abstractions at different levels of expressivity. Some of them (frequently aimed at providing system governance for real world scenarios) tackle this issue by interpreting events as symbolic facts rather than limiting them to pure numerical metrics. In other words, they move the events from the subsymbolic world of arithmetic to the symbolic world of logic.

Among those approaches, the most relevant regarding this work is *Normative Systems*. Normative Systems focus on the concepts of *Norm* and *Normative Environment* (also known as *Instution*) in order to create normative frameworks. These frameworks will be able to guide and restrict the behaviour of (intelligent software) agents. The core idea behind these concepts is that the interactions among a group of agents (either between them or with the environment around them) are to be ruled by a set of explicit norms. Such norms are expressed in a computational language representation that agents can interpret and understand.

To put our proposal in context, in the following sections we will provide a brief introduction to the following relevant research topics:

- Agents and agent societies
- Organisations,
- Institutions
- Norms and Norm-Governed Systems
- Governance in dynamic normative contexts

## 1.1 Agents and Agent societies

Computational systems are steadily moving from being based on individual stand-alone computational resources to a situation where distributed, open and dynamic systems play an important role. Agent based systems [WJ95] [Wei99a] are an alternative for designing and implementing open and dynamic systems. Nowadays, agent systems are being used in an increasingly wide variety of applications, from relatively small systems (such as e-mail filters for spam detection) to large, open critical systems (such as air traffic control). In order to be autonomous, effectively reducing the need of user intervention, software agents are able to get some information about the world in which they operate. This way, they can solve most of the minor problems they find during their operation cycle. This allows Multi-Agent Systems to operate in scenarios where typical software solutions might have problems operating, for instance, scenarios where the environment is not predictable (*e.g.*, due to the actions performed by entities external to the system that alter it). Typically, agents as autonomous intelligent entities have provided a solution to deal with complex systems that have multiple and distinct components.

As defined by Wooldridge and Jennings in [WJ95]:

> *An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives.*

The important aspect on this definition is autonomy, which means that agents can operate on their own, without the need for human guidance. An autonomous agent has full control over its internal state and its actions, which means that an agent can decide weather or not to perform a requested action. The definition also situates the agent in a particular environment, which can be both sensed and affected by the agent. In other words, agents are capable of both perceiving the state of the environment and altering it. The definition states that agents have goals to achieve and show flexible and pro-active behaviour towards them. Agents are also capable of social behaviour, they can communicate, compete and cooperate among them. Last but not least, agents also show the capability to learn as they interact with the environment and with another agents. This set of capabilities, which are commonly shown by intelligent beings, is what has lead some researches to refer to agents as intelligent agents.

Being capable of social behaviour, the idea of building agent systems which capture notions from human society is recurrent in Artificial Intelligence research. The notion of agent societies is associated to two main motivations:

- Being aware of the benefits of human societies, building software systems which incorporate them.
- Integrating already existing agent systems with rapidly growing of electronically-based human societies associated with the growth of the Internet.

The main idea behind a society is to allow its members to coexist in a shared environment and pursue their respective goals in cooperation or competition with others. Therefore, artificial social systems [MT95, ST95] define an abstract social level over computational systems. The social level models the Multi-Agent System as a society of entities, defining structured patterns of behaviour that facilitate and enhance the coordination of agent activities [VS03]. The social structure is usually defined by the following elements: roles, interaction rules, norms and a communication language. Both rules and norms define the desired behaviour of members in the society. They are established and enforced by institutions that will provide legitimacy and security to society members. By specifying the expected (*i.e.*, accepted) patterns of behaviour, guidelines are imposed on the patterns of interaction among agents, facilitating them. At the same time this will effectively reduce the danger of a combinatorial explosion in the number of available patterns of interaction.

When designing agent societies the constraints the society imposes to an agent's actions might vary. Davidsson [Dav00] analyses the types of constraints in the context of artificial societies. This allows him to produce a taxonomy able to classify the four different approaches to the development of agent societies. The approaches are as follows:

- Open agent societies, where anyone with an Internet access may contribute one or more agents without any particular restrictions.
- Closed agent societies, where an agent based approach is adopted by a team of software developers for implementing a complex software system.
- Semi-closed agent societies, where anybody may contribute an agent. However, entrance to the society is restricted and behaviour might be monitored by an Institution. Agents are implemented and run locally.
- Semi-open agent societies, just like semi-open agent societies, but agents are implemented and run on remote servers.

## 1.2 Organisations

With the aim of achieving adaptability to the environment, organisational models that specify the structure of societies have appeared during the last years, playing an important role on the design of information systems. In [Les90] Lesser provides this definition of organisation:

> *An organisation provides a framework for activity and interaction through the definition of roles, behavioural expectations and authority relationships (e.g., control).*

In [WJK00a] Wooldridge, Jennings and Kinny propose another definition:

> *We view an organisation as a collection of roles, that stand in certain relationships to one another, and that take part in systematic institutionalised patterns of interactions with other roles.*

From these definitions, the following main features of an organisation are inferred:

- An organisation is formed by agents (*i.e.*, individuals) that manifest a particular behaviour.
- The overall organisation might be divided into partitions. Such partitions might overlap. We will refer to these partitions as groups.
- Introduction of the concept of role: Behaviours of agents are functionally related to the overall organisation activity.
- Agents engage into patterns of activities (that is, dynamic relationships) which might be classified using a taxonomy of roles, tasks and protocols.
- The different types of behaviours are related through relationships between roles, tasks and protocols.

A very important element in organisations is the concept of role. A role can be defined as the description of an abstract behaviour of agents. A role defines the constraints an agent has to satisfy in order to obtain it, the benefits the agent will receive when playing it and the responsibilities associated to that role. The role is also the placeholder for the description of patterns of interaction agents playing that role will have to perform.

An organisation might be divided in two aspects, the structural aspect and the dynamic aspect [FGM04]. The structural aspect consists in a partitioning structure and a role structure. The partitioning structure defines how agents are assembled into groups and how groups relate to each other. Then, for each group, a role structure is defined, including the set of roles and their relationships. The dynamic aspect of the organisation defines the following elements:

- The modalities to create, dissolve, enter groups and play roles.
- How these modalities are applied and how deontic elements (obligations and permissions) are controlled.
- How both partitioning and role structures are related to the agent's behaviour.

## 1.3 Institutions

Institutions are another type of social structure which can be used to define and check for acceptable behaviour. Institutions are identified by the set of social constraints that govern both the behaviour and the relationships between members of a society.

### 1.3.1 Human Institutions

North [Nor90] claims that institutional constraints ease human interaction by shaping choices and making outcomes predictable. Via these constraints, institutions can become more complex while keeping reduced interaction costs. Constraints also allow participants on the institution to act (and expect others to act) according to a list of fixed protocols of interaction. Institutions can create trust among parties, even when they do not have much information about each other.

According to North, institutions can be created from scratch and remain *static* or be continuously *evolving*. Institutions can be informal (*i.e.*, defined by informal constraints such as social conventions and codes of behaviour) or formal (*i.e.*, defined by formal rules, political and judicial, economic rules or contracts). The purpose of formal rules is to promote certain kinds of interaction while increasing the cost of undesired kinds of interaction.

### 1.3.2 Electronic Institutions

Just like in human institutions, in Multi-Agent Systems the lack of information about other members of an organisation leads to low trust, which can penalize the efficiency of interactions on the organisation. An Electronic Institution [VS03] is the model of an human institution with a norm specification provided in some machine-readable formalism. The idea is capturing the essence of an institution (norms and protocols) in a machine processable form. Institutions, as norm providers and enforcers, try to solve the following issues in the context of Multi-Agent Systems:

- Reduce the uncertainty about other agents' behaviours inside the institution.
- Reduce misunderstanding with a common set of norms governing the interactions inside the institution.

- Allow agents to predict the outcome of a certain interaction between the participants in the institution.
- Simplify the decision making process inside the agents participating in the institution by reducing the number of possible actions in each particular context.

## 1.4 Norms and Norm-Governed Agents

In scenarios where Multi-Agent Systems are applied to systems with an overall holistic goal, it is not desirable that an agent's autonomous and emergent behaviour diverges from the overall goal of the system. In order to limit this agent autonomy and ensure a certain coherence between the goals of the particular agents and the overall goals of the system, agent organisations are designed. The institutional interpretation of reality and the rules of behaviour for the agents within an organisation are described using norms.

Normative Systems are composed of regulative and non-regulative components:

- Regulative rules explain what should happen ideally (*e.g.*, a winner of an auction will pay for the auctioned product before leaving the auction). Regulative rules are typically represented in terms of deontic concepts [GAVSD06] such as obligations, permissions and prohibitions. However, some authors [MBB$^+$11, Sin96] use other notions such as authorisations or commitments.
- Constitutive rules (also known as counts-as rules) describe *'what counts as what'* in a given institution. In order words, they are able to transform low level facts perceived from the environment into high-level interpreted institutional facts. It is important to note that constitutive rules are relative to a context. As a clear example, raising a hand in the context of a Japanese auction counts as leaving the auction, effectively refusing to bid for the product. However, raising a hand in an English auction counts as making a bid for the product.

Regarding how this set of norms affects agent behaviour, two main lines of though arise [GAD07]. On the one hand, some authors [GCNRA05] see norms as inflexible restrictions to agent's behaviour. Agents will never be able to violate them. On the other hand some authors [Ald07] see norms as a guide to agent's choices that agents can choose to follow or not. Anyway, both lines of thought agree on one fact: norms make the behaviour of the agents more predictable, effectively reducing the complexity of the system.

In [UBSA10] Artikis *et al.* define the social state of a Norm Governed system as an extension of the physical state with social attributes (*e.g.*, sanctions imposed on participants, norms violated, etc.).

A Normative Multi-Agent System is a set of autonomous agents that must comply to social norms [BvdTV08, Dig99]. Normative Multi-Agent Systems are typically used to design electronic institutions. Even though Normative Multi-Agent Systems implementations are different from one another, the following general characteristics can be identified [LLLd07]:

- Membership: Agents in a society must be able to deal with norms, but they must also be able to recognise themselves as part of the system. This social identification means agents adopt the norms of the society, and by doing so, show their willingness to comply with these norms.
- Social Pressure: Effective authority can not be exerted if penalties or incentives are not applied when norms are violated or complied with respectively. This control must not be an agent's arbitrary decision, it must be socially accepted.

- Dynamism. Normative Systems are dynamic by nature. New norms are created and obsolete norms are abolished. Compliance or non-compliance with norms might activate other norms and force other agents to react accordingly.

### 1.5 Governance in dynamic normative contexts

The word *governance* derives from a Greek verb that means '*to steer*' and implies assuring that an organisation provides a good pattern of results while avoiding an undesirable pattern of bad circumstances. When governance is applied to *Normative Systems* results and circumstances are instantiated into states of the normative environment. Therefore, governance on *Normative Systems* can be seen as checking patterns of states of the normative environment.

Therefore, one of the requirements when implementing *Normative Systems* is being able to assess, at runtime, the state of the normative environment (*e.g.*, a norm has been violated, a sanction has been fulfilled, etc.). Some existing lines of research (*e.g.*, [ÁNAVSD10] [OPVS$^+$09]) already try to tackle this issue on some scenarios.

Human legal systems are seldom static. They evolve through time as regulations change [Fri77] [Hud93], effectively adapting to new situations and behaviours. Typically regulations will change due to a wide range of factors, some examples are:

- Behavioural changes: Changes in the behaviour or the distribution of the actors forming the institution. If the institution has an increase on free-raiders or misbehaving actors the regulations and protocols governing it will have to evolve to become more strict, in order to detect and sanction the non-compliant behaviours. However, if the actors in the institution are well-behaved the regulations can be relaxed and minimised in order to improve efficiency.
- Technological advances: New technologies (*e.g.,* electronic cigarettes, e-prescriptions, *etc.* ) will typically imply an update on regulations and protocols.
- Social changes: Habits that were widely accepted in the past and are not socially accepted anymore. For instance, slavery.
- Environmental changes: Actors actions can alter the environment, and therefore change the set of norms regulating the institution. For example, the pollution of a water source will imply a change in the regulations and protocols governing the actors that access the water source.
- Contextual changes: Different institutions have different regulations and protocols. An actor in transition between different institutions will see a change in the regulations and protocols that apply to him. For instance, a driver travelling through different countries will find different speed limits along his way.

Therefore, the normative context (that defines the normative environment) will not be static in many scenarios. It is adapted, and expands and contracts as new norms are added to the institution and removed from it respectively. As electronic institutions are meant to mirror the regulations and protocols governing the institution they represent, it is easy to foresee that some of the electronic normative environments will not be static. Therefore, we will see electronic normative environments evolving through time as norms are inserted, removed and updated.

Under this condition, a monitoring system must be able to continue computing the state of the normative environment at runtime. This is an important feature, because in many application domains we can not afford to perform the changes on the normative context off-line, as we could be missing some important information *w.r.t.* the normative contexts. For instance we could be missing a massive amount of norm violations, in case miss-behaved actors use the off-line period to do their will without being sanctioned. Furthermore, it must be guaranteed that the monitoring system can keep producing states of the normative environment that are consistent with the changes performed on the normative context. For instance, if a norm has been removed from the normative context, it does not make sense anymore to compute normative states where the norm has been violated.

Existing frameworks tend to fail when trying to apply them to dynamic scenarios, where the set of regulations and protocols governing the system will evolve over time. In this thesis we aim to tackle this problem, enriching and extending existing lines of research.

## 1.6 Objectives

The main objective of this thesis is to formalize and develop an extended normative framework and architecture to cope with scenarios where the normative context is dynamic, therefore it can expand and contract at runtime. We also aim to meet some requirements on the normative context update operations:

- The operations are to be performed at runtime, without having to stop computing the normative state. If we stop observing the actors in the institution to update the norms, we could be missing some important information *w.r.t.* the normative state. For instance, some actors might take advantage of the off-line period and violate norms without being sanctioned.
- The normative state computed must be consistent with the expansion and contraction operations. That is, it might not make sense to account for norm violations that happened before the norm was added to the normative environment. If we remove a norm, we might have to forget about former norm violations.
- Following the prior requirement, we want our system to be expressive, effectively providing tools for the legislator or the policy maker to decide how to perform the updates on the normative context. If the legislator wants to add a norm and account for norm violations that happened even before the norm was added, we want to open this possibility. If the legislator wants to remove a norm, and forget about past norm violations, compensating for the fines agents paid for violating it, we want to provide this option.

Our goal is to tackle this issue by formally defining the operations to be supported in order to allow for expanding and contracting the normative context. Then, we take a base formal framework for norm monitoring and extend it, mainly providing support for a more expressive norm life cycle.

By fulfilling these objectives, we provide the following contributions:

1. A conceptualization of the extended framework supporting the context update operations.
2. A formal model of the extended framework, focused, but not limited to, the context update operations.

However, we do not want to stop on the formalization process. Our goal is to implement a running prototype providing expansion and contraction operations. By providing the prototype we achieve the following contributions:

1. The model of an architecture for a normative monitor supporting the context update operations.
2. A prototype for a norm visualization component for documenting the results of our research.

Finally, our goal is also to demonstrate our framework by applying it to two use cases: e-health systems and wastewater management on a river basin. By applying the framework to the use cases, we achieve the following contributions:

1. A model of the agents involved in wastewater management in the river basin. The model includes both the social and the normative structure.
2. An instantiation of the generic architecture to the river basin scenario.
3. A model of the agents involved in the e-health scenario. The model includes both the social and the normative structure.
4. An instantiation of the generic architecture to the e-health scenario.

## 1.7 Structure of this document

The rest of the document is structured in five chapters and one annex.

*Chapter* 2 provides an analysis of the state of the art. It mainly covers work on existing frameworks for computing the normative state of a system. It also includes analysis on works regarding Normative Systems and existing works on dynamic normative frameworks. Finally, an analysis of Normative Systems applied to environmental management and healthcare is provided.

*Chapter* 3 introduces NoMoDEI, our framework for computing normative states. The framework effectively allows for performing expansion and contraction operations on a normative context at runtime, that is, without having to stop the monitoring process. The chapter starts by introducing an example to motivate norm dynamics. The chapter goes on by formalising a base framework for monitoring normative contexts. Then, the base framework is extended by introducing norm dynamics. Norm dynamics are introduced via the expansion and contraction operations to be supported in order to allow monitoring dynamic normative contexts. The chapter provides a formal representation of these operations based on a formal extension of the base framework. The main extension is defining a more expressive norm life-cycle. Then we proceed to provide a particular instantiation of our formal framework. The chapter goes on by providing the formal algorithms for supporting expansion and contraction operations. Special care is put into ensuring the normative states computed after any operation are consistent. All the algorithms proposed on this section are formal and generic, ready to be applied to any particular instantiation (*i.e.*, implementation) of the formal method for computing normative states we use as basis. Then, the chapter provides an architectural design that includes the definition of the different components in the architecture including interfaces for norm update, monitoring components (with special emphasis on the monitor's knowledge base) and institutional agents. The architectural definition is abstract enough so as to be able to instantiate it with a variety of technical solutions. Then, the chapter instantiates the formal algorithms for supporting expansion and contraction operations, providing implementations details. The chapter also provides an instantiation of the architecture, with particular technological solutions selected for the different components. Finally, conclusions are drawn.

*Chapter* 4 provides an application scenario based on wastewater management in a river basin. The scenario covers all aspects of the formalisation and implementation of the framework, and a justification of the advantages provided by it on a scenario where norms can simplify operational management, but where adaptation to changing and unexpected situations requires using the framework presented in this document. The chapter provides a wide example of our framework, covering all the normative components it provides and all the possible operations on norms it supports.

*Chapter* 5 provides an application scenario based on e-health. The scenario covers all aspects of the formalisation and implementation of the framework, and a justification of the advantages provided by it on a highly regulated scenario, but where adaptation to changing technological and social contexts requires using the framework presented in this document. The chapter provides interesting examples of the application of our framework, such as the option of using sanctions as rewards, the utility of retroactive promulgation of norms and the usage of constitutive norms as an abstraction layer.

Then, *Chapter* 6 summarizes this document, including conclusions drawn from each chapter. The chapter includes a small section justifying the relevance of the work presented to Artificial Intelligence. It includes an analysis on the accomplishment of the objectives pursued by the thesis. Finally, it presents future lines of research.

Finally, *Annex* A contains an in-depth analysis of two lines of work for norm monitoring that have influenced the framework presented in *Chapter* 3.

# CHAPTER 2

# State of the art

This chapter contains an analysis of the relevant state of the art in the scope of our proposal. It covers both, state of the art for the framework that we propose and state of the art for the applications of our framework to real world scenarios.

The first two sections of the chapter (§2.1 and §2.2) are devoted to relevant works in the literature that can be applied to any domain. They are works directly related to the framework we propose in this PhD thesis. The first two sections introduce some relevant works on Norms and norm monitoring systems on §2.1. They go on with a deep analysis of two formal frameworks for norm monitoring in §2.1.6 and §2.1.7. The next step is §2.2 an analysis on support for Dynamic Normative Contexts.

The following two sections of the chapter (§2.3 and §2.4) introduce domain specific works applied to environmental management and healthcare. They are relevant for the application of the framework we propose to our test scenarios. In §2.3 we introduce an analysis on Normative Systems for environmental management scenarios in general and wastewater treatment scenarios in particular. The next section §2.4 introduces an analysis on Normative Systems applied to health care scenarios.

Finally, discussion on the analysis of the state of the art is provided in §2.5.

## 2.1 Norms and monitoring systems

One of the requirements when implementing *Normative Systems* is being able to assess, at runtime, the state of the normative environment. Although there are many works in the literature that refer to Norms and Nomative Systems, this section will focus on works presenting relevant information regarding norm monitoring.

### 2.1.1 Social Power and Norms

In her PhD thesis Fabiola López y López [LL03] studies the impact of social power and norms on agent behaviour. In this work, among other things, she performs an interesting analysis of norm dynamics. She states that norms are not a static concept, there are several processes started by norms in which different agents can be involved. The state of a norm can be inferred from these processes.

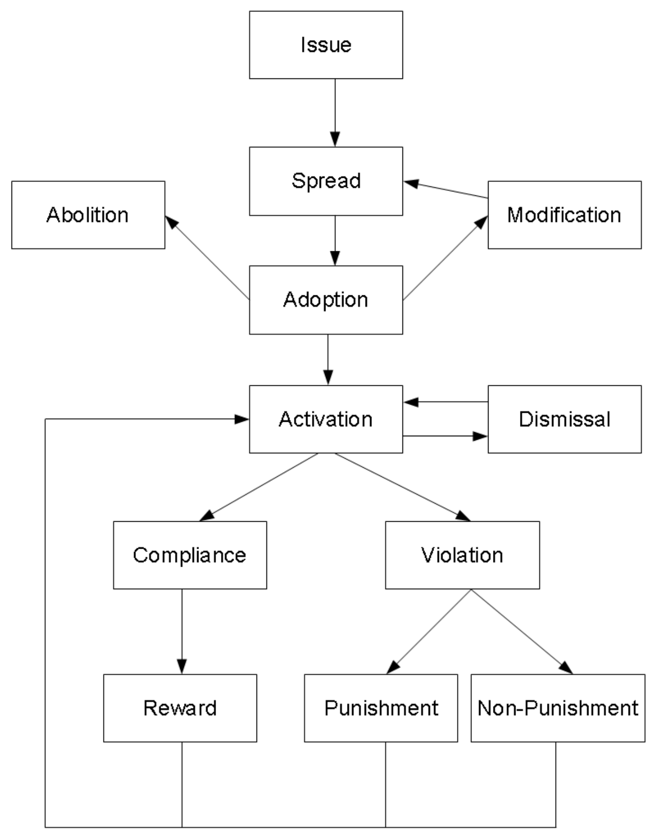



Figure 2.1: Norm transitions in Lopez y Lopez's Norm Life Cycle

Indeed one of the most interesting concepts defined in her work, regarding the scope of this proposal, is the definition of the transitions between one state of the norm and the following state. Such transitions form the norm life-cycle, and are depicted in *Figure* 2.1 and are defined as follows:

- **Issue**: Legislators issue a norm. Legislators are institutional agents, that is, either agents connected to a graphical interface and controlled by users or autonomous software agents introducing norms in the system autonomously.
- **Spread**: The norm is spread among the participants in the institution via direct or indirect communication mechanisms.
- Adoption: Adoption of the norm by addressee agents takes place. By adoption, agents express its willingness to fulfill the norm as a way of being part of the society.

- **Activation**: Once a norm has been adopted it remains inactive until its applicability condition is satisfied.
- **Dismissal**: Exception states can be defined where agents are not obliged to comply with some norms (norms in dismissal state) and therefore, such norms can be ignored.
- **Compliance**: Whenever addressee agents comply with a norm, it reaches a compliance state.
- **Reward**: After a norm is complied with, a reward can be offered. However, since agents responsible for the application of rewards have limited perception it is possible that the compliance of a norm remains unnoticed and, therefore, no reward is offered even when the design of the norm states a reward is to be offered if the norm is complied with.
- **Violation**: Whenever addressee agents violate a norm, it reaches a violation state.
- **Punishment**: After a norm is violated, offenders are punished.
- **Non-punishment**: Since agents responsible for the application of punishments have limited perception it is possible that the violation of a norm remains unnoticed and, therefore, offenders are not punished.
- **Modification**: Legislators modify a norm. The norm goes to spread state and the modification is spread among the participants in the institution via direct or indirect communication mechanisms.
- **Abolition**: Legislators abolish a norm, effectively removing its legal effects. The norm goes to spread state and the abolition is spread among the participants in the institution via direct or indirect communication mechanisms.

In general, the definition of norm transitions presented in Fabiola's work seems expressive and rich. The idea of adopting such norm transition definition in the work we propose looks promising.

However Lopez y Lopez's work focuses on studying impact of social power and norms on agent behaviour, without providing means to monitor and observe this behaviour, effectively putting it in contrast with the norms.

### 2.1.2 Opera and Opera+

In [Dig04] [DWX02] Dignum *et al.* present **Opera**, a framework and methodology for agent societies that specifies the steps to design and develop an agent-based system in the scope of a particular domain. The methodology takes the organisational perspective as a starting point, effectively accounting for the influence of the social organisation model on the functionality and objectives of an agent society. The idea is creating a general framework based on an organisational view that can complement and be complemented by existing agent-based design methodologies. This will effectively contribute to the acceptance of agent-based solutions by real world organisations and give answer to the following development challenges, as introduced in [Syc98]:

- How to engineer Multi-Agent Systems that are practical and applicable to real world scenarios?
- How to decompose complex problems into simple tasks and allocate these tasks to individual agents?
- How to coordinate agent control and communication?
- How to coordinate agent capabilities so agents act in a coherent manner?
- How to include other agents and the state of the coordination into agent's reasoning process?

- How to detect and solve conflicting goals between collaborating agents?

The motivation of taking the agent paradigm [Wei99b] as starting point is that it provides a natural and easy way to view and characterise intelligent systems. As intelligence is deeply coupled with social interactions, Multi-Agent Systems provide insights about the interactions between intelligent beings, as they organise themselves into social structures (*e.g.*, groups, societies, *etc.*) in order to achieve their individual and common objectives. In general, Multi-Agent Systems represent the interactions between agents and are the virtual counterpart of societies and organisations typically found in the real world. Therefore, the methodology proposed in **Opera** simplifies the design of the system by reducing the distance between the electronic organisation and the real world organisation it models.

Previous works, tackle the aspect of coordination in Multi-Agent Systems by considering technical aspects related to coordination and planning. The **Opera** methodology goes further by taking into account the implications of the coordination model for the Multi-Agent System architecture and design method.

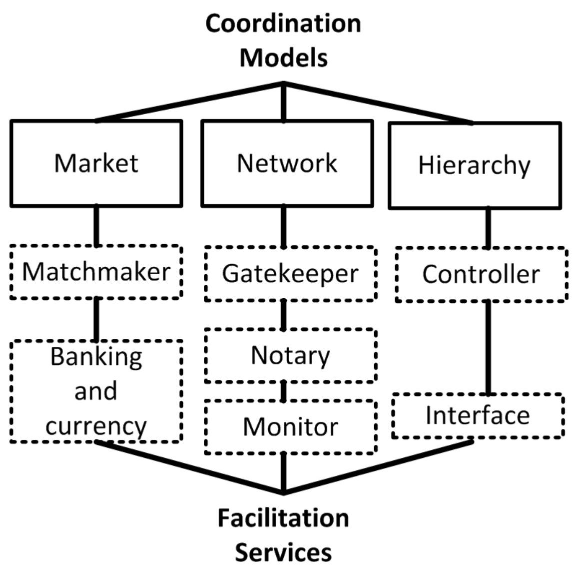



Figure 2.2: Opera coordination models and facilitation services

The coordination model depends on the topology of the organisation represented by the Multi-Agent System. An organisation is a specific solution created by autonomous actors to achieve common goals. Social interaction emerges from a set of negotiated social norms and is regulated by mechanisms of social control. The way a society organises and balances its objectives is effectively reflected in the coordination model. For instance, when transaction costs are high (*e.g.*, uncertainty or unpredictability of events, transactions require specific or expensive investments, the risk of opportunistic behaviour of patterns is high, *etc.*) societies tend to choose a hierarchical coordination model, tightly controlling the transaction process. However, if transactions costs are low (*e.g.*, straightforward and non-repetitive) market coordination models are the optimal choice. Finally, if transactions are highly dependent on the different actors of the organisation and coupled to coordination rules and norms, a network coordination model is the optimal choice. *Figure* 2.2 shows a classification of the different coordination models along with the facilitation services supporting them.

In markets, agents are self-interested, following their own goals and preferring freedom of association over security and trust. Facilitation services offered by the institution are limited to matchmaking. Matchmakers keep track of agents in the system, knowing their needs (demand) and capabilities (supply) and mediate in the process of matching demand and supply of services. Matchmakers can also provide reputation mechanisms by building confidence to their customers and offering some guarantees to the agents using them. In case ways to value the services exchanged are required (*e.g.,* for determining profit and fairness) banking and currency facilities can also be provided.

Network organisations are built around general patterns of interaction known as contracts. Coordination between agents is highly dependent on clear communication patterns and social norms. Agents in network societies are willing to trade some freedom for enhanced security and trust. Facilitation services include gatekeepers, notaries and monitoring agents. Gatekeepers accept and introduce new agents into the market, informing them about the capabilities of the market and negotiating the terms of a social contract between the new agent and the members of the market. Notaries keep track of collaboration contracts between the agents, mediating in disputes. Monitoring agents (typically consisting in trusted third parties) check agent actions (typically by looking at their states) and impose sanctions if undesirable states of the world are reached (*e.g.,* an agent not paying for a service).

Hierarchical organisations are typically composed of altruist agents willing to contribute to a common global goal. Facilitation services assume the global control of the society and mediate between the agents and the world outside the institution. For this, controller and interface agents are defined. Controllers monitor and orient the overall performance of the organisation, coordinating the actions of the autonomous agents (whose perspective and actions are typically local) and aligning them with holistic goals. Interface agents regulate the communication between the agents in the institution and the outside world.

The **Opera** methodology defines the following levels, providing increasing refinement of the resulting system into a more structured and precise form:

- Coordination: determines a coordination structure for the domain and designs a coordination model based on market, network or hierarchical architectures. *Table* 2.1 summarizes the models available and their properties.
- Environment: based on the coordination model selected in the previous step, describes the interaction between the organisation and the environment where it operates. Includes a description of the expected functionality of the society, that is, what the society is intended to do or produce as output.
- Behaviour: describes the intended behaviour of the organisation in terms of roles and agent interaction patterns. Is concerned only with a high level description of agents and their goals.
- Agent: introduces the internal architecture of the agents in terms of requirements for communication, action, interface and reasoning. Provides agents in the organisation with the capability of performing their own task and interacting between them.

| | **Market** | **Network** | **Hierarchy** |
|---|---|---|---|
| **Society Goal** | Service Exchange | Collaboration | Service production |
| **Agent Autonomy** | High | Medium | Low |
| **Agent Goals** | Individual | Both | Global |
| **Relation forms** | Negotiation Protocol | Defined by society norms | Fixed workflow |
| **Communication Capability of agents** | Based on standards | Defined by society norms | Fixed on design |
| **Interface to environment** | Open for agents with identification | Admittance procedure | Closed for agents, open for data via interface agents |

Table 2.1: Coordination Models

The **Opera**+ methodology [JDT12] extends **Opera** for describing collaboration relationships in inter-organizational partnerships. Inter-organizational collaboration will typically take place in complex, dynamic and unpredictable environments. If agents require to analyse the overall setup and decide on their participation, regulating structures should be represented explicitly. Furthermore, they should be represented independently from the acting components and at different levels of abstraction. The idea of the **Opera**+ methodology is to specify different types of components in a common model creating the opacity inter-organizational systems require, and at the same time making it easier for actors to understand their partnerships.

Inter-organizational interactions are represented in two dimensions. The specification dimension depicts the regulating structures in terms of connected roles and organisations. It focuses on the objectives to be achieved in inter-organizational collaboration. The enactment dimension presents the acting components in terms of agents enacting the roles. It focuses on which role achieves what objectives.

The **Opera**+ methodology refines roles into atomic and composite roles. Each composite role refers to a unique organization at a lower level in the hierarchy which elaborates the objective of the composite role into fine-grained roles. This refinement will effectively provide more information and define more constraints on how to accomplish the objectives. Atomic roles are not further specified, providing an abstraction layer over agent enactment. Therefore, an atomic role can be enacted by any type of agent, whereas a composite role presents two main options:

1. Be directly enacted by a composite agent, as long as the internal organisation of the agent matches the one of the composite role.
2. Be indirectly enacted by a set of independent agents, where each of them is enacting a sub-role.

Regarding the relation between agents and roles, each agent can enact one or more roles, as long as the capabilities of the agent meet the requirements of the role. Furthermore, an agent enacting a role is allowed to further extend the role specification according to its own requirements and functionalities, which may not be completely known in advance.

Finally, a particular specification can cover multiple role enactments. For each enactment, a set of agents enacting the role in the specification is defined. Some of the agents may have their own internal understanding of the objectives of the role they enact. Therefore, agents can effectively extend the inherited specification according to their own capabilities. This allows agents to further refine the specification to better achieve their objectives. Furthermore, it also provides a balance between autonomy and conformity.

The **Opera** methodology provides support for normative structures (specially focused on network coordination models) via monitoring agents. Market coordination models also support normative frameworks via contracts and negotiation protocols. Hierarchical models lack a normative framework due to the their tightly controlled nature, where agents trade autonomy for trust and security. Being a generic methodology, **Opera** does not provide details about the structure or architecture of these normative frameworks. However, the methodology does not specify support for norm dynamics. Regarding the **Opera**+ methodology, it focuses only on structural aspects, leaving the normative issues for future work.

We consider our approach can benefit from the **Opera** methodology in order to define the social structure underlying to our normative structure (*e.g.,* roles affected by the norms) and the **Opera** methodology can be effectively extended with a dynamic normative framework for supporting evolving institutions, that change as the topology of the agent society evolves when groups of agents enter or leave the organisation. For instance, we can have a initial set of agents aiming for autonomy on a market coordination model and evolve to a hierarchical model as the society changes when new (non-trustworthy) agents enter the system. Both the method to trigger a change in the coordination model and how to support this change are out of the scope of this document, but we can effectively provide a method to make the normative framework present in the methodology evolve when social changes happen.

### 2.1.3 Autonomy vs. Conformity

In his PhD thesis [Ald07] Huib Aldewereld presents a scenario where autonomous agents decide weather or not share information (privacy sensitive information) based on the applicability of local norms. This scenario arises the need for a global frame of enforcement for the global norms. In other words, local agents (each of them bound to local procedures and rules) have to adhere to global regulations. It has to be checked that information transactions are not in conflict with the global laws, and this check has to be performed on real-time. Most software and even agent design methodologies see these global regulations as extra requirements on the analysis phase. The result: the regulations are hard-coded into the software or the agents themselves. Therefore, if the regulations change, it becomes very hard to track all the changes required in the implementation, as there is no explicit representation of the regulations (*e.g.*, in form of norms).

The alternative is to have an explicit representation of norms. That is, an electronic institution. As introduced in §1.3 an electronic institution is an entity defining a set of norms over the individuals participating in the institution. Electronic institutions provide a safe environment effectively mediating the interactions of the individuals participating in it. Such mediation is achieved by expressing the expected behaviour of agents by means of an explicit specification of the norms. However, according to Aldewereld, introducing electronic institutions in highly regulated domains usually requires to solve issues related to:

- Abstractness of human regulations.
- Lack of operational information.
- Implementation of norm enforcement from an institutional perspective.

In his thesis, Aldewereld tackles these problems by introducing a framework to make the connection between the norms and the agent practice explicit. Both the normative specification and the procedure (*i.e.*, protocols that agents can use to achieve common tasks) of an electronic institution are derived from the laws and regulations that govern the domain. Some form of enforcement is required to ensure none of the agents participating in the electronic institution break the laws and regulations that govern the domain. Aldewereld proposes two options for performing this enforcement:

- Restricting the agents to a specific set of procedures that is known to be norm-compliant.
- Monitoring the behaviour of the agents and punishing them in case they violate a norm.

Aldewereld argues the later holds more benefits since restricting the agents to pre-defined behaviours restricts their autonomy and only enables them to act via procedures defined beforehand. This effectively negates the agent's capacity to handle new and unforeseen situations. However, this requires the implementation of an active norm enforcement system based on the detection of violations, and reaction to such violations. This can be hard because of the abstractness of the norms and the lack of operational information in them. Therefore, Aldewereld tries to tackle this issue by introducing a formal representation of norms based on a representation in deontic logic. Annotations including all the operational information necessary to implement the norms from an institutional perspective are included.

A semi-automatic procedure for translating the specifications given by the norms into particular patterns that can be used to create interaction protocols is provided. This is because, introducing protocols for an institution provides agents with a default manner to achieve certain common tasks without having to take the norms into notion (as they know beforehand the protocols are norm-compliant). The procedure is achieved by introducing an intermediate level of landmarks between the norms and the practice where the landmarks express the important immediate steps any protocol should contain. This way, agents know that following a protocol, by the letter, they will never break any norm.

By putting together active norm enforcement and the design of norm-compliant protocols for electronic institutions, Aldewereld is able to create a framework that links laws (expressed in a normative specification) to the agent practice. The framework takes the first steps needed in order to cover the most important aspects of institutional design. Aldewereld assess the implementation of norms, as proposed in his framework by the creation of an active norm enforcement procedure, ensures a good balance between autonomy and conformity of the agents participating in the electronic institution.

However, the framework does not propose mechanisms for norm dynamics. Even if the framework is able to cope with normative changes by dropping the current institution and creating a new one (with the new set of norms) it is unable to cope with changes in the norms seamlessly. When changing norms the transition between the old institution and the new one may cause some consistency problems in the set of norms, for instance norm violations may be lost when transitioning between institutions.

### 2.1.4 Designing Invisible Handcuffs

In his PhD thesis [Gro07] Davide Grossi develops a precise view of institutions and organisations and how events can be interpreted to conform an institutional reality. Grossi states that institutions are Normative Systems, and Normative Systems impose terminologies which are defined by means of constitutive rules. Normative Systems define contexts (sets of situations that can be ordered from concrete to abstract) which make the rules of the Normative System true. The idea is providing a way to make institutions and organisations formal and therefore *visible* in some way.

In his work, Grossi analyses the differences between constitutive and regulative norms.

- Counts-as statements are statements talking about institutions, which are viewed as terminologies defining contexts. Counts-as rules can express:
  - What logically forms a given terminology (classificatory counts-as).
  - What logically follows from a given terminology which does not hold in general (proper classificatory count-as).
  - Which axioms does the terminology contain (constitutive counts-as).
- Regulative rules can be seen as special types of constitutive rules. They define the concept of violation for a given institution.

Grossi defines organisations as structures laid upon the set of roles of the organisation. The structure has a direct impact on the activities of the agents enacting the roles of the organisation. In particular, links between roles define the following information:

- Stablish what activities can be performed by the agents, in the sense links between roles define what kind of system transitions are possible and under which conditions in a given Multi-Agents System.
- Specify what kind of effects can be determined by the performance of certain activities by the agents enacting the roles at issue. Such effects can be institutional (therefore they depend on the being in force of the set of constitutive rules) or mentalistic (certain speech acts that, when successfully performed, will change the mental state of the receiver).
- Formal properties of the structure formed by the link between roles can effectively derive performance, robustness, flexibility and efficiency metrics.

In his work, Grossi already outlines the necessity to tackle the issue of norm implementantion from a game-theoric perspective (by using implementation theory and mechanism design). The idea is to:

- Understand how a set rules can be implemented on a society of agents via appropriate mechanisms.
- Evaluate the impact of different sets of rules implemented by different mechanisms on the same society.

This would effectively provide norm designers with tools to better understand the impact of sets of norms on the to-be-regulated society, effectively improving the quality of the legislative action.

However, there are neither implementations nor tools supporting Grossi's language and mechanisms.

### 2.1.5 Efficient Norm Monitoring Frameworks

This subsection analyses some proposals for improving the efficiency of production systems and norm monitoring systems.

In [UBSA10] Visara Urovi, Stefano Bromuri, Kostas Stathis and Alexander Artikis present a formal framework with an associate run-time support infrastructure that is able to compute physically possible and permitted actions at each point of time. The framework is also able to apply sanctions that should be applied to violations of prohibitions. In order to provide run-time support, a specific version of Event Calculus with support for efficient temporal reasoning is selected. The framework can be distributed in order to support real-time computation of the physical and social states. The main idea is describing a Multi-Agent System as two concurrent and inter-connected composite structures that evolve over time. One represents the physical environment and the other represents the social environment. In order to be able to compute both of them at run-time, the framework is able to distribute the environments in different computational resources.

The framework presented in this work is exemplified using an open packet world problem. In such problem, agents are located in a grid-like world (*e.g.*, formed by an $8 \times 8$ grid) and pick coloured packets, delivering them on the delivery posts of the same colour. As agents have a limited view of the grid, they need to collaborate in order to solve the problem in an efficient way. Such collaboration is achieved by agents placing flags in particular locations. Such flags let other agents know that a particular area has been explored and, therefore, has no packets left. Agents compete among them, as points are assigned to every agent based on the number of packets he has delivered. Therefore, cheating agents might try to obstruct other agents by placing flags in non-explored areas, fooling other agents into thinking there are no more packets left in the area. As these actions result in a loss of efficiency of the society of agents (one can easily see the average time to pick all the packets increases if fake flags are placed) norms are introduced in order to prevent agents from using these techniques. Whenever an agent puts a flag in an unexplored area, his points are reduced.

The usage of norms implies a computational process must be used for detecting cheating agents and applying the corresponding sanctions. This approach fulfils this requirement by implementing a component known as *Social Calculator*. This component encapsulates the state of the physical environment, and access it in order to check for violations. Then, extends the state of the physical environment by storing detected violations, effectively building up a normative environment. The social calculator can also answer agent queries about their permissions at a specific point of time. The idea is distributing the state of a particular environment into sub-states. Each of them will have its particular computational resource supporting a local social calculator component. For instance, a $8 \times 8$ grid representing the world can be divided into four $4 \times 4$ grids. Procedures for sharing the normative environment on *border* areas are described in the approach as well. Finally some performance tests are executed to show that distributing the normative state allows for larger-scale (in this case, with more agents) Multi-Agent Systems with norms.

In [Che93] Albert Mo King Cheng explores parallel execution as an approach to achieve higher execution speeds in rule-based systems. This approach is specially suited for domains requiring high performance and real-time response. When rule-based systems are used to monitor and control real-time systems the ability to meet stringent response times constraints is as important as the ability to produce correct results. In particular, this approach focuses on demonstrating how rule-firing parallelism can be automatically extracted from a rule-based system by analysing the source code. Rule-firing parallelism consists in firing non interfering rules in parallel. Cheng applies his approach to the *EQL* rule firing language, which contains rules with the typical left-hand side and right-hand side format.

Cheng's approach is based on building a *High Level Dependency Graph* from the set of rules contained in the source code of the program. Cheng formally ensures the process of building the graph can be performed in a bounded amount of time. Cheng starts by defining the concept of rule dependency. Formally:

**Definition 1** ***(Rule Dependency)***

*Given two disjoint set of rules $S$ and $Q$, a predicate for querying the elements in the right part of the rule $R$ and the analogue predicate for the left part of the rule $L$:*

*$S$ is independent from $Q$ if the following conditions hold:*

- $L_S \cup L_Q = \emptyset$
- *Rules in $Q$ do not potentially enable rules in $S$. That is: $R_S \cup L_Q = \emptyset$*

□

Once the concept of rule dependency has been formalized, Cheng models the set of rules as a graph. Every rule is a node. There is an edge between two nodes if the rules representing them are not independent as defined above. Then, a *Strongly Connected Component* analysis is performed, finding the *Strongly Connected Components* (denoted as *SCC* from now on) in the system. Please, notice this process has effectively created clusters of dependent rules. The *High Level Dependency Graph* is the graph containing as nodes the different *Strongly Connected Components* detected and edges among nodes whenever rules in the components are dependent. Finally, for deciding whether two rules can be fired in parallel or not, the *High Level Dependency Graph* can be effectively queried for checking whether two given norms belong to the *SCC* or not.

In [GFK+88] Anoop Gupta *et al.* explore very fine-grained parallelism for achieving significant speed-ups on the OPS5 production system implementation. Gupta states that when trying to perform such optimizations one must be very careful with resulting overheads, otherwise, overheads might nullify the gain achieved by the optimization. The optimization is based on a Rete Match algorithm [For82], a data-flow network able to significantly speed-up the match phase on production systems. The idea is, first exploiting the fact that only a small fraction of the working memory changes every cycle by storing results from match in previous cycles and reusing them in future cycles. Then exploiting the similarity between condition elements of production rules to effectively reduce the number of tests that have to be performed during a matching phase.

In order to support the parallelism, Gupta defines a control process and N match processes. The control process is responsible for conflict resolution, handling input/output and evaluating the right-hand side of the rules. It also starts the match process and computes working memory changes with respect to the last execution cycle. As soon as the first working memory change is detected, information about that change is passed to the match processes that start to work. When the last match process finishes, the control process performs conflict resolution, and is ready to start the next cycle. Regarding the match process, the idea is divide the match in independently schedulable units of work known as tasks and execute them in parallel.

In [LE80] Lesser and Erman present a new model for organizing distributed systems. In this model, the distributed system is able to function effectively even if particular processing nodes have inconsistent or incomplete views of the information required for their computation. Lesser and Erman apply their model to the area of interpretation systems, that accept a set of simple signals from an environment and produce higher level descriptions of the events in the environment. The idea is developing an architecture able to allocate processing resources at the sensor sites requiring only limited communication between the processors. It is proposed as a solution able to provide real-time response and reliability on environments requiring the use of limited communication bandwidth.

Their main idea is to search for a general solution by the incremental aggregation of partial solutions. The aggregation process is able to effectively solve errors and uncertainty derived from inconsistent or incomplete local views of the information required for solving the problem.

Lesser and Erman apply their model to the speech understanding software Hearsay-II [EHRLR80]. Each computational node will be a completely functional Hearsay-II system, but will have access only to a particular segment of the speech input data of the utterance. Neighbouring nodes cooperatively generate an interpretation of the complete utterance by sharing partial interpretations based on their local views.

The Hearsay-II system allows for interpretations to be built by combining partial interpretations derived from different knowledge modules known as *knowledge sources*. The architecture's design allows for both collaborative and competitive problem solving among the different knowledge sources. First the set of all possible partial interpretations defines a search space. The more alternative hypothesis generated by the different knowledge sources, the larger the fraction of the space that is searched. Then, a subset of the existing partial interpretations is selected for extension. At this step, the resulting extended partial interpretations compete for selection with those previously generated. On the one hand the system tries to focus quickly on information that constraints the search in order to prevent combinatoric explosions of the search space. On the other hand, promising tentative decisions can be made and reused later when new information is available. In general, three requirements must be satisfied in order to effectively use this approach for problem solving:

- Sufficiency of knowledge: Knowledge able to generate some sequences of partial interpretations that culminate in a correct complete interpretation.
- Sufficiency of credibility evaluation: The credibility function that chooses what partial interpretation to expand must be able to rate higher correct complete interpretations than incorrect complete interpretations.
- Sufficiency of control strategy: The system must be able to find a correct complete interpretation within the bounds of the computational resources allocated to the task.

When applying his model for distributed systems to the Hearsay-II system Lesser ends-up with a system with the following characteristics:

- There is a network of systems (network nodes), and each of them is able to perform significant local processing in a self directed way. For instance, if a node does not receive a given piece of information in a limited amount of time, it is able to continue processing, using whatever information is available to it.
- The parts of the problem a particular node is responsible for working on is known as *area of interest*. It is defined by the information it needs and produces. Areas of interest in different nodes overlap. Therefore, the local knowledge base of a node might be inconsistent with respect to the knowledge bases of neighbouring nodes. Nodes, however, are able to resolve the uncertainty in this information using an iterative asynchronous interchange of partial tentative results.

- Control of cooperation among the nodes is decentralized and implicit in the autonomous behaviours of the individual nodes. Each node will use its local estimation of the state of the problem for controlling its processing (*i.e.*, which fraction of the local search space to expand) and the transmission of information to another nodes.

As the overall solution is constructed via incremental aggregation (putting together mutually consistent information) incorrect partial solutions are automatically discarded during this process, as they will naturally contain non consistent information. This will effectively allow the distributed system to reduce the impact of incorrect decisions caused by incomplete and inconsistent local information. Finally, defining problem solving as a search process implies exploring many alternative partial solutions. These searches can be effectively carried out in parallel in different network nodes. This parallelism implies there are multiple paths from where the solution can be derived. So, it is possible to over-pass errors that would be considered as fatal errors in conventional problem solving systems (*e.g*, process getting stuck in local maxima).

### 2.1.6 Towards a Formalisation of Electronic Contracting Environments

One line of work regarding norm monitoring which is of special interest for our proposal is the one presented by Nir Oren *et al.* [OPVS$^+$09] when defining a formal representation for electronic contracts. This section provides a brief summary on Oren's approach. A more detailed description can be found in §A.1.

In their work Oren *et al.* present a formal representation of contracts that focuses on the specification of clauses by a model of norms. The model supports both constitutive and declarative[1] norms. Within this model, a norm is associated with a status, that changes as the environment (either the physical environment or the normative one) changes. A normative environment is defined. It can be used to track the status of a set of norms through their life cycle and to describe predicates to evaluate norm's status. Such predicates are used by agents to evaluate and reason about the status of norms. Furthermore, the predicates can be used by agents to predict how their actions are going to affect the normative environment. The applicability of their framework to real world domains is proved by showing an example where the execution of a contract, taken from a real world problem, is monitored.

Oren *et al.* motivate their work on the appearance of web-services and the need to regulate interactions between them. According to them, this fact highlights the desirability of fully automated contracting. Fully automated contracting formalization requires the following properties:

- Ability to describe the contract on a machine interpretable way
- Ability to describe the contract on a form over which inference may be performed
- Techniques for automatically generating and enforcing contracts
- Protocols allowing agents to create and modify contracts

Oren *et al.* define contracts as normative documents, able to impose a set of requirements on agent behaviour. Such requirements come in the form of actions the agent may undertake and states of the world (*i.e.*, environment) the agent should or should not allow to occur. Normative components are formalized based on deontic logic, with a special focus on tracking the changing state of norms. For instance, tracking when the norm is active (that is, the norm has a normative force on an agent) and when the norm has been violated.

[1] Oren's declarative norms correspond to the regulative norms in our framework, as we will introduce in §3.3

It must be remarked that in this later case, they consider a violated norm can be *unviolated*. At the same time, They define norms as socially derived prescriptions specifying some sets of agents (known as the norm's *targets*) must perform some action or see to it that some state of the world is met. Norms are imposed on the target by some entity (known as norm's *imposer*). *Imposer* is granted (typically by the society) some power to impose the norm by imposing penalties on agents violating a norm. Oren *et al.* impose the following restrictions on the contracting domain in which he operates:

- The model should allow for determining if a violation took place. If it did, which agent was responsible for causing the violation.
- Support for norm verification. That is, determining weather conflicts between norms occur, and weather a norm could be complied with (never, sometimes or always).
- Agents should be able to use the normative model to support their reasoning process, effectively deciding which action they should undertake.
- Norms must be able to cope with conditions based on the state of another norms, and specially, norm violations.
- The model must be extensible, allowing for different knowledge representations and reasoning mechanisms to make use of it.

Oren *et al.* consider norms to have normative force only in particular situations. Norms that do not apply to a situation are considered *abstract* and are *instantiated* when their *activation condition* holds. An *instantiated* norm remains active until a specific *expiration condition* holds true, then the norm does not have normative force any longer. They distinguish between two different norm types, *obligations* and *permissions*. Finally, it defines a *normative goal*, used to specify when the norm is violated (in the case of obligations) or what the agent is allowed to do (in the case on permissions). Please note that additional norm types can be created from the original norm types. For instance, a prohibition is an obligation with a negated normative goal.

However, the framework does not introduce mechanisms to update the set of clauses in a contract. Intuitively in case the clauses in a contract are updated, the actual contract can be removed, adding a new contract with the modified clauses. However, in order to apply this mechanism, procedures to ensure the consistency of the normative environment must be applied. For instance, what happens if the old contract has been violated? Do violations hold in the new one? Lacking these procedures, Oren's approach does not fit perfectly in scenarios where contracts can evolve and be modified over time.

### 2.1.7 Normative Monitoring: Semantics and Implementation

One of the lines of work regarding norm monitoring which is of special relevance for our proposal is the one presented by Sergio Alvarez-Napagao *et al.* [ÁNAVSD10]. They present a formalism for the monitoring of regulative (deontic) and substantive (constitutive) norms based on *Structural Operational Semantics* and a reduction to *Production Systems* semantics. This section provides a brief summary on Alvarez-Napagao *et al.* approach. A more detailed description can be found in §A.2.

Alvarez-Napagao *et al.* state that literature on the topic of Normative Systems is growing rapidly, and many approaches are being presented. However, most of these approaches lack a proper implementation of the ontological connection between brute events and institutional facts.

They justify the need of mechanisms where normative specifications can be added to the agent's knowledge base at run-time and be practically used in the agents' reasoning.

This would allow agents to be able to interpret institutional facts from brute ones (by using constitutive norms to decide if a given brute fact counts as a given institutional fact on a particular context) and to decide what ought to be done (by using regulative norms to change the normative environment from an unacceptable state to a an acceptable one).

Alvarez-Napagao *et al.* propose using production systems to build a norm monitoring mechanism that can be used by agents to perceive the normative state of their environment. The same system can be used by the environment (*e.g.*, in the form of institutional agents) to detect non desirable states of the world (*e.g.*, norm violations) and enforce actions that will return the world to a normal state (*e.g.*, sanctions and repair actions). Their basic idea is that an agent can configure, at a practical level, the production system at runtime abstract organisational specifications and sets of counts-as rules. Basically, in their approach, the detection of normative states is a passive procedure consisting in monitoring past events and checking them against a set of active norms. By using a forward-chaining rule engine, events will automatically trigger the normative state without requiring a design on *how to do it* .

One of the most interesting features in Alvarez-Napagao *et al.* approach is being able to decouple the normative state monitoring from the agent reasoning. This allows for easily implementing third party and facilitator agents that are capable of observing, monitoring and reporting normative state change or even enforcing behaviour in the organisation. For this to be achieved the following elements are required:

- A direct syntactic translation from norms to rules.
- A logic implemented in an engine that is consistent with the process to be accomplished.

However, the monitoring mechanism does not account for norm change. In case norms are updated, the current monitoring process can be stopped and a new monitor started, accounting for the new set of norms. However, it may cause the system to stop observing the social reality while the monitor transition is performed, and in some scenarios we can not afford to stop observing the social reality even for a brief amount of time. For instance, if we take this approach, miss-behaving actors may take advantage of the situation and violate norms massively, as the system is transitioning between monitors and therefore unable to observe these norm violations and sanction them. Furthermore, this procedure would require communication between the old and the new monitor for communicating the normative state (*e.g.,* which norms have been violated) and techniques to perform this exchange of information have not been introduced in Alvarez-Napagao's approach. Therefore, even if the approach is suited for static normative contexts it must be extended in order to support dynamic ones.

## 2.2 Dynamic Normative Contexts

Just like in human legal systems, it is easy to foresee that electronic normative environments must not be static, but will have to change and evolve through time, as regulations change to adapt to new situations and behaviours. Furthermore, as seen in §1.4, dynamism is one of the characteristics of Normative Systems.

That is one of the main reasons why there is already some work regarding dynamic normative contexts. This section will analyse some of the most relevant approaches regarding the scope of this proposal.

### 2.2.1 Changing Legal Systems: Abrogation and Annulment

In [GR08a, GR08b] Governatori *et al.* explore alternatives for modelling abrogation and annulment operations on Normative Systems. Their work aims at contributing to create a model for capturing norm change in Normative Systems by taking some steps in this direction: exploring the notion of legal modification. Legal modification is the way in which human law systems implement norm dynamics. Governatori *et al.* focus on explicit law modifications that modify the system by specifying why and how existing norms should be modified. To do so, they start by studying the concepts of *derogation* and *annulment*:

- Annulment: Makes a norm invalid and removes it from the legal system. It applies *ex tunc* effectively preventing the norm to produce its legal effects, no matter if such effects where produced before the annulment operation or not. For instance if someone is imprisoned as a sanction for violating an annuled norm, he/she should be set free.
- Abrogation: Makes a norm invalid and removes it from the legal system. It applies *ex nunc*. Prevents the norm from producing its legal effects from now on, but does not alter legal effects produced before the abrogation of the norm. For instance if someone is imprisoned as a sanction for violating an abrogated norm, he/she should remain imprisoned.

In their work, Governatori *et al.* try to tackle the formal modelling of abrogation and annulment by using Defeasible Logic [PN88, ABGM01] because rule-based systems seem a natural way to represent legal systems. Their idea is trying to adjust belief and theory revision in Defeasible Logic in order to capture both abrogation and annulment. They explore the following options on their first approach:

- Revising extensions of Normative Systems. Consists in blocking the effects of abrogated or annulled rules. Requires a method to state which rule generates which effects. This approach is problematic if several rules generate the same effect, and only one of them is abrogated or annulled.
- Adding exceptions. New rules (with higher priorities) are introduced in order to block the effects of abrogated or annulled rules. This approach does not reflect how the law implements norm changes. Legal effects of rules can guide how rules can be changed but are unable to specifically determine what and how rules are changed.
- Revision of normative bases. Consisting in removing some rules from the base of a theory. This approach is problematic when dealing with *ex nunc* modifications as the whole rule should not be removed.

In general, Governatori *et al.* conclude that both norm abrogation and annulment can only be partially represented in the approaches he explores. Therefore, he tries to address the issue from a different starting point, via *Temporal Defeasible Logic* [GRS05]. Temporal Defeasible Logic is an extension of Defeasible Logic where every literal has an associated timestamp. The main idea behind this second approach is capturing dynamics of a legal system as time-series of its versions. Every time a modification is applied to the legal system, a version (norm repository) is stored. If a modification is applied to the legal system, the subsequent norm repository will contain a modification of the rule, preventing it from entailing consequences. Governatori proposes introducing new auxiliary literals to signal whether literals are revoked as a consequence of an annulment or not. The idea is creating copies of the annuled norms using these auxiliary literals that will never entail consequences. However, as *ex nunc* operations only must prevent this entailing since a particular point in time, Governatori proposes leaving a set of norm repositories unmodified (the ones corresponding to the point of time before abrogation). These norm repositories will be able to effectively entail the consequences of the rule.

However Governatori's approach presents the following limitations:

1. Norm *annulment* presents a problem under this approach, conclusions of *annulled* norms might remain on the repository after the norm has been *annulled*, and the solution proposed to remove the conclusion seems quite ad-hoc.
2. Governatori's solution provides no explicit support for *retroactive promulgation*.
3. Governatori's approach is not able to update obligations and permissions, in fact Governatori states that an explicit differentiation between norms, obligations and permissions has to be made.
4. Governatori's approach does not provide support for *constitutive norms* (*i.e.*, counts-as) or institutional powers.
5. Governatori approach is mainly formal. He does not mention any implementations or tools supporting his norm change mechanisms. There is no outline of the feasibility to create them or the computational cost of his operations.

### 2.2.2 Dynamic Context Logic and its Application to Norm Change

In [AGHL09] Aucher *et al.* formalize a dynamic logic that supports characterizing operations of contraction and expansion theories. Then, they use the logic to develop and axiomatic and semantic analysis of norm change in normative contexts. The work focuses on two specific context change operations:

- Context expansion, accounting for norm promulgation. Formally $X + \psi$. That is, we enlarge a context with some worlds that satisfy $\psi$.
- Context contraction, accounting for norm derogation. Formally $X - \psi$. That is, we enlarge a context with some worlds that satisfy $\neg\psi$. An exception to this is the case where $\neg\psi$ already exists in the context ($\neg\psi \in X$). In this case $X$ is not modified. Please note that this fact implies there might be several possible contractions of a given model $X$.

Then, Aucher *et al.* use this context (model) expansion and contraction operations to formalize norm change. Formally:

- Rule promulgation: Given a classificatory rule $\varphi \to \psi$, and a context $X$ the rule is added to the context. Formally: $X + (\varphi \to \psi)$.
- Rule derogation: Given a classificatory rule $\varphi \to \psi$, and a context $X$ the rule is removed from the context. Formally: $X - (\varphi \to \psi)$.
- Obligation promulgation: Given a state of the world $\psi$, an atom that holds true when there is a violation $\mathcal{V}$ and a context $X$, adding the obligation of $\psi$ to the context implies there is a violation when $\neg\psi$ holds. Formally: $X + (\neg\psi \to \mathcal{V})$.
- Obligation derogation: Given a state of the world $\psi$, an atom that holds true when there is a violation $\mathcal{V}$ and a context $X$, removing the obligation of $\psi$ to the context implies there is no more a violation when $\neg\psi$ holds. Formally: $X - (\neg\psi \to \mathcal{V})$.

It is important to note how rules entailing violations are added and removed from the system when obligations are added or removed respectively.

However Aucher's approach does not provide support for both *ex tunc* and *ex nunc* forms on norm operations. Only one normative context expansion and one normative context contraction operations are introduced on his approach; both operations seem to (implicitly) be of *ex nunc* type. Furthermore, Aucher's approach is mainly formal. Implemented frameworks or tools supporting his proposal are not mentioned. Finally, even though classificatory rules are supported there is no support for institutional power.

## 2.3 Normative Systems for Environmental Management

There is a growing interest on wastewater treatment. On the one hand, demand for wastewater treatment is increasing fast [GBRHP12], and will continue to do so in the near future. On the other hand, wastewater management is a complex scenario requiring the simultaneous consideration of technical, economic, social and environmental factors. This is due to two main reasons:

- Even though there are many process-specific technologies capable of adequately treating wastewater, there is no single technology or even group of technologies capable of providing a global solution to the potentially infinite number of wate-water scenarios. Typically, different modules are combined and coordinated in a treatment train to meet specific requirements. The increase of wastewater treatment processes available has caused an explosion in the number of possible combinations for a treatment train configuration.
- The need to coordinate a big number of activities performed by different actors with different goals, sometimes not aligned with the holistic goals of the system, also makes this scenario a complex one. Not to mention the fact that wastewater treatment presents several situations (*e.g.*, heavy rains, droughts, river pollution) that require a response from the system in a timely manner.

The combination of this two factors results in the fact that wastewater management scenarios are very complex. Normative systems are one of the solutions we can apply to tame this complexity.

This section presents a short survey on the existing work in the area of Multi-Agent Systems applied to environmental scenarios, with special emphasis on the works focused on water management and on applying Normative Systems to align individual and social goals.

### 2.3.1 Distributive Justice for Self-Organised Common-Pool Resource Management

In [PBM14] Jeremy Pitt *et al.* present an approach for enduring common pool resource management. The approach is based on complementing Ostrom's institutional design principles [Ost90] with Rescher's theory of distributive justice [Res66] based on the concept of legitimate claims.

On the one hand, Ostrom identifies eight institutional design principles for enduring common-pool resource management, effectively defining a general framework to specify institutions enabling appropriators to sustain a resource. Summarizing, the most relevant principles *w.r.t.* the approach presented are:

1. Congruence of the resource allocation method taking into account both the resources available and the environment.
2. Participation of the actors affected by the resource allocation method in the selection of the method.

On the other hand, Rescher identifies a set of mechanisms for allocating resources. According to him, distributive justice consists on identifying legitimate claims in the context, accommodating them in case of plurality and reconciling them in case of conflict. The idea is using fairness mechanisms to keep the different actors as happy and satisfied as possible. This will effectively avoid situations where unsatisfied actors start misbehaving, and by extension the negative effects of missbehaviour on the society.

A fair resource allocation method providing a fair resource allocation outcome requires operational choice rules grounded on a theory of distributive justice. Therefore, Pitt *et al.*

propose to complement Ostrom's institutional design principles with Rescher's theory of distributive justice. The main idea is developing a formal model of resource allocation according to legitimate claims. The model is implemented in a Multi-Agent System based simulation that combines the principles of enduring self-organising institutions [JAP13] with methods for distributive justice.

The simulation consists in a variant of the linear public good game (LGP) [Gäc07] known as LGP'. The LGP game is a cooperation game used to study voluntary contributions, opportunities for free riding and many other issues. In the game, a set of agents form a cluster. The game is played in rounds. At every round, a particular actor in the cluster will perform the following actions:

1. Determine the resources it has available.
2. Determine its needs for resources
3. Make a demand for resources.
4. Make a provision for resources.
5. Receive an allocation for resources.
6. Make an appropriation of the allocated resources.

The LGP game makes the following assumptions:

- There is full disclosure among the actors participating in the scenario.
- There are no monitoring costs, that is, we can have an omniscient module observing actor's actions, interactions and mental states.
- There is no cheating in appropriation. If a particular actor is allocated an amount $m$ of resources, the actor will appropriate an amount $n$ of resources, where $m \geqslant n$.
- There are no diminishing returns.

The assumptions have to be relaxed in open systems (*i.e.* there might be cheating in appropriation as we do not have controls on actor's behaviour) and in systems modelling an economy of scarcity with a limited pool of resources. In order to meet this requirements, the LGP' game variant is defined [PS12]. In such variant actor's need for resources is greater than the resources it can generate by itself. Therefore, actors depend on each other (*i.e.* they have to cooperate) and there is an incentive not to comply with the rules. Furthermore, in LGP' games actors are organized in clusters with access to a common pool of resources. If an actor considers resource distribution is not fair, it will leave the cluster. Therefore, cluster cohesion (how long a cluster can withstand without its member abandoning it) can be used a metric of fairness in resource distribution. This scenario makes it necessary to design a mechanism that will:

- Put an incentive on the provision of resources.
- Encourage an accurate representation of resource requirements.
- Discourage excess appropriation (*i.e.* appropriating more resources than the ones allocated).

The nature of LGP' results in a scenario where actors demand more resources than they actually need, provision less than what they actually generate and appropriate more than the amount they are allocated. Applying the fair resource allocation method proposed (that results of the combination of Ostrom's and Rescher's approaches) ensures fairness in demand and provisioning, effectively providing an enduring cluster where no actor is encouraged to leave due to a non-fair distribution of resources. However, appropriation presents a real threat for cluster's endurance. By ignoring their allocation, cheating actors bypass the whole legitimate claims mechanism designed for a fair allocation of resources. This is because, even if allocation is fair, the fact actors can cheat on appropriation (effectively bypassing allocation) results in an unfair distribution of resources. Authors claim that, in order to tackle this issue, they need methods for preventing this behaviour, such as using retributive justice to punish cheating actors. Authors propose to use monitoring and sanctioning mechanisms to identify non-compliant behaviours (*e.g.,* appropriating more resources than the ones allocated) and fine cheating actors.

### 2.3.2 Improving urban wastewater management through an auction-based management of discharges

In [MBD+11] Murillo *et al.* propose the use of auction based processes in which Wastewater Treatment Plant (WWTP) capacity is sold in order to coordinate different industrial water discharges in the scope of an Urban Wastewater System (UWS). The coordination is aimed at providing a steady inflow rate of wastewater and pollutants to the WWTPs, avoiding situations where the plant receives peaks of wastewater and pollutants which can not be properly treated.

The treatment capacity of a plant is limited, therefore all pollutants arriving at the WWTP should be below certain limits, otherwise the water can not be properly treated and there is an environmental impact on the river. Current solutions tackle this goal by imposing a fixed amount of authorized discharges to each industry. Such solutions might not be sufficient to guarantee the correct treatment of wastewater, because, even though individual industries will respect WWPT capacity thresholds, simultaneous industrial discharges are not taken into account. Therefore, WWTP limits can be exceeded even if every single industry connected to the plant has not bypassed the limits. In order to tackle this issue authors propose an auction process in which the resource capacity of the WWTP is sold. The approach has already been applied to the energy market, in which energy from different sources is auctioned in order to favour the use of non-pollutant sources [HZ00] as well as for auctioning $CO_2$ emission credits [Vig07].

The problem tackled in [MBD+11] could be solved using a centralized approach. Given all the planed discharges for industries a new schedule is generated for each of them, such that the maximum capacity of the plant is never exceeded. This technique is known as pre-emptive cumulative scheduling [BLPN99]. However, it presents two main drawbacks:

1. Finding the solution is computationally expensive and often not applicable in practice.
2. A centralized scheduler would take all the decisions using complete information from industries. However, industries may not be willing to reveal information about their production process (specially taking into account other industries are involved in the process) and this information is mandatory to compute their wastewater treatment requirements.

In order to tackle the problem, and bypass these drawbacks, authors propose a coordination mechanism where both hydraulic and pollutant treatment capacities are modelled as individual resources shared by all industries. Each time a conflict in a resource is detected (either hydraulic capacity is exceeded or the capacity to treat a particular pollutant is exceeded) an auction is held in order to determine which of the conflicting discharges will be authorized and which one will be delayed. In order to support this mechanism it is assumed every industry has a retention tank where it can store a discharge whenever it is delayed and empty it later on. It is also assumed every industry can estimate in advance the discharges it will generate according to its production process.

During the coordination process proposed, every industry agent will communicate the WWTP agent its discharge schedule for a given period of time (*e.g.,* a day). The schedule contains the discharges planned for a given period of time, and for every discharge, information about water quantity and pollutant concentration. Once the WWTP agent has received all the schedules for a given day it starts checking for conflicts. When a conflict is detected. the involved industry agents are informed about it, and an auction is started to solve it, effectively forcing some industries to modify their discharge schedule. Rescheduling should not alter industry agents' production process. In case it has to reschedule the agent will try to store the rejected discharge into its retention tank. The discharge of the tank is scheduled as the first action to be performed by the agent once the conflict has been solved. Please note that if tank is already full (or there is no tank) the discharge will be performed anyway. Authors propose a mechanism [MMB$^+$07] to minimize these situations during the auction process, but to the best of our knowledge, it is not applied. Furthermore, authors assume industries are naive and they will not try to cheat. They state in the real world application a mechanism to ensure the bid price corresponds to the industry's urgency for discharging should be implemented.

Experimental results performed via simulations demonstrate the method proposed by the authors effectively reduces the impact of industrial discharges. This is achieved by reducing the variability in the quality of the waters entering the treatment plant. A steady concentration of water quantity and pollutant concentration allows the plant to process pollutants more efficiently.

### 2.3.3 Ant Colony Optimization-based Method for Managing IndustrialInfluents in Wastewater Systems

In [VCP12] Verdaguer *et al.* propose a system aimed at avoiding temporal overloads (caused either by water volumes or pollutant concentrations) that exceed a Wastewater Treatment Plant (WWTP) capacity. The idea is using a combinatorial optimization procedure with multiple constraints that is applied when the plant lacks the capacity to treat the wastewater in all of its influents. A Multi-Agent System manages all the information related to the state of the entire system, effectively applying optimal influent assignment criteria via an ant-colony optimization based method.

Industrial wastewater has a high variability regarding quantity and pollutant concentration. WWTPS in general, and in particular the ones using biological treatments, require to smooth both flow and loads to protect the microbial community used to treat wastewater. Current solutions[AAKS98] tackle this goal by imposing a fixed limit of authorized discharges to each industry. Such solutions might not be sufficient to pursue this two objectives:

- Ensure the optimal capacity of the plant at any time.
- Treat industrial effluents according to their urgency.

In the scenario presented in [VCP12] industry agents with water retention tanks are connected to (*i.e.* are influents of) a WWTP agent that treats industrial wastewater before discharging it to an efluent, typically a river. When industrial wastewater is generated via their production process, industries can choose weather to discharge it to the WWTP or store it on the retention tank, which has limited capacity. The proposal is using a coordinating agent that aims at finding a good combination of industrial discharges *w.r.t.* WWTP efficiency, that is, as much capacity as possible is used from the plant without overloading it. The process of looking for a good solution is performed using ants randomly placed on a graph-like search space, where nodes are industrial activities and edges possible discharges.

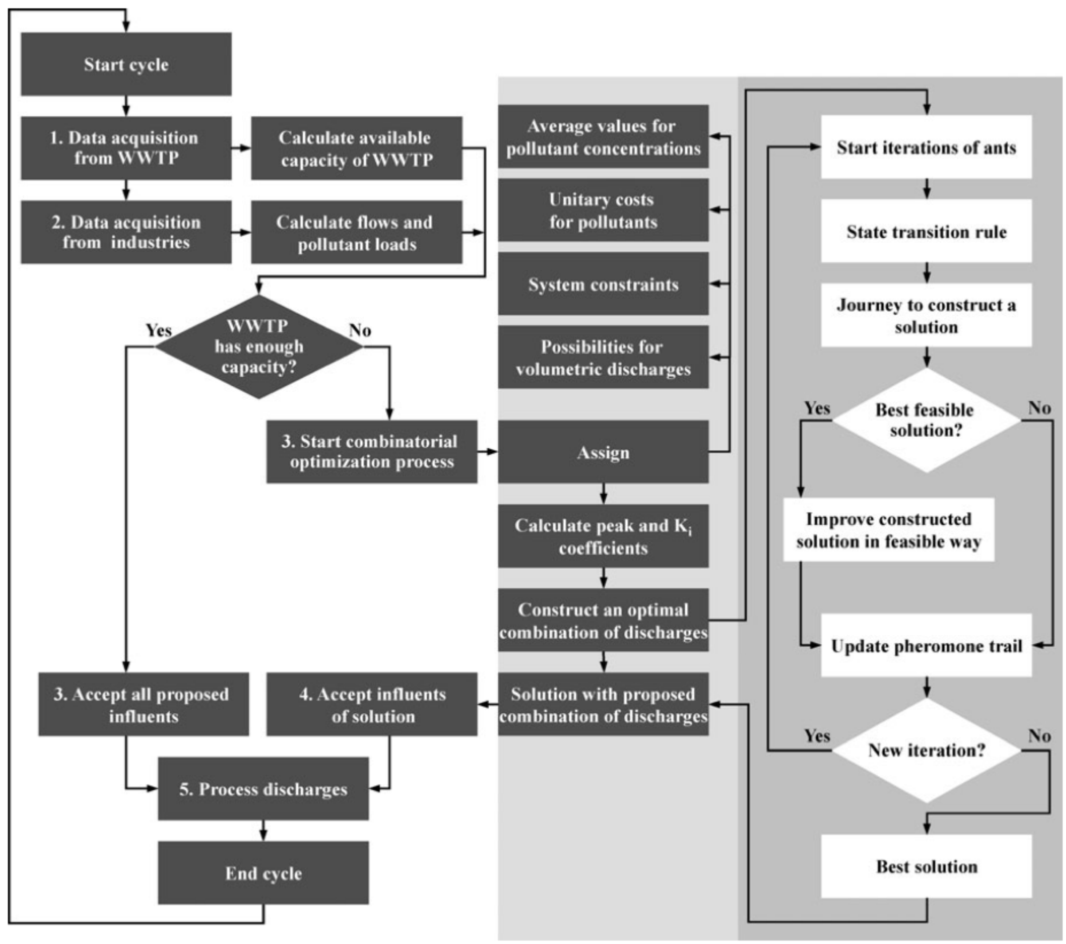



Figure 2.3: Ant Colony: decision cycle of the coordinating agent

Figure 2.3 shows the decision cycle of the coordinating agent used in the proposal by Verdaguer *et al.* The section in white background depicts the rules used for linking authorized flows to industrial activities. The rules are as follows:

1. Data acquisition from the WWTP agent. This step determines the current status of the WWTP.
2. Data acquisition from the industry agents. Where industrial activities on the next 24 hours are planed. Industrial wastewater flow and pollutant compositions are derived and communicated.
3. In case WWTP agent does not have enough capacity to treat all industrial wastewaters, a combinatorial optimization process for discharges is executed. Otherwise, industrial discharges are authorized.
4. Accepts industrial discharges as authorized in step 3.
5. Process industrial discharges as accepted in step 4.

The sections in gray background cover the combinatorial optimization process, which consists in iterations of two main phases. In the first one each ant is placed in a random location of the solution space and constructs a solution using a probabilistic state transition rule. Ants that cover a path leading to a successful solution reinforce their pheromone trail. The idea is generating large amounts of pheromone when a successful solution is built, and smaller amounts in the case the solution is not successful. Therefore, the amount of pheromone in a particular path constructing a solution is related to the quality of the solution, and the best solution found in the iteration can be identified. In the second phase the best solution is selected. The solution is improved first using an optimization technique aimed at promoting the partial discharge of wastewaters stored in retention tanks. If a new (*i.e.* higher) volume of discharge provides an improvement over a feasible solution, the new volume (coming from tank discharge) is added and the solution considered. Then, the solution is improved using a second optimization that performs a neighbourhood search. The optimization increases one of the pollutants, reduces another and leaves the rest unchanged. Pollutants are selected at random. If a new pollutant concentration provides an improvement over a feasible solution, the concentration is updated and the solution considered.

The method proposed results in a tendency to favour large pollutant concentration discharges in the case industrial activities have the same value regarding water volumes. When WWTP influents show large fluctuations in pollutant concentrations, the method proposed in [VCP12] reduces variability effectively achieving a better wastewater treatment cost for the solution.

### 2.3.4 Integration of freshwater environmental policies and wastewater treatment plant management

In [CAGP13] Coromines *et al.* present a simulation based analysis of the different directives for regulating the characteristics of the discharged water and the chemical characteristics of the received ecosystems.

In the last years political awareness of river quality issues has steadily grown and therefore legislation is adapting accordingly. In the case of the European Union, where Wastewater Treatment Plants (WWTPs) have been identified as major sources of point source pollution [BRT06], two different directives regulate both the characteristics of the discharged water and the chemical characteristics of the received ecosystems. Current legislation reflected in WWTP management implies plants will adjust their operation, effectively adjusting waste removal capacities. This is because the legislation sets a limit on the pollutant concentration of the receiving water body, which depends on the pollutant concentration of both the WWTP and the receiving water body. Therefore, the impact on receiving waters as regulated by the legislation changes over time as a result of river-flow variation. The negative impact of this policy is worsened in arid and semi-arid regions with low river flow values, such as the Mediterranean area, because the difference between the amount of water discharged by the WWTP and the river flow is relatively high. In fact, in some extreme cases the river flow will be nearly empty unless it receives discharged waters from the WWTPs.

With this scenario in mind, authors suggest considering the specific characteristics of the water bodies in the management of the WWTPs in order to minimize the impact on receiving water bodies and achieve the objectives of good environmental status. They also propose an economic analysis estimating the volume, prices and costs of the water services in order to evaluate the cost effectiveness of the alternative possible regulations. Coromines *et al.* model a particular WWTP located in the north east of Spain [DCT$^+$09] and run several simulations to find the combination of plant operational settings that match the target as specified in the legislation. Optimizations are applied separately for winter and summer seasons, effectively taking into account contextual information such as temperature and rainfall.

The simulations show it is possible to integrate the chemical status of receiving water bodies into WWTP management. Therefore, there is a gap in the current wastewater treatment legislation, which should be updated to account for an integrated perspective allowing for a more flexible management of the WWTP, effectively maximizing not only the ecological benefits of the system, also the social and economical ones. Simulations show the environmental impact can be decreased to guarantee the achievement of good chemical status while maintaining similar treatment costs in the WWTP. This is achieved by adjusting plant operation conditions to the pollution loads measured not only in the influent, but also in the river.

### 2.3.5 $m$Water, a Case Study for Modeling Virtual Markets

In [GGBN13] Garrido *et al.* present $m$Water, a regulated virtual market where autonomous agents trade rights for the use of water in a closed basin. The aim is achieving the complex balance between economic, administrative, environmental and social factors involved in water management. The hypothesis is more efficient uses of water may be achieved inside an institutional framework where water rights may be exchanged more freely under different market conditions [Tho97]. However, there are many aspects that may be regulated with many parameters involved. The consequences of the many combinations available are difficult to predict, not to mention the conflicting interests of the different actors participating in the institution. Therefore, policy makers need tools that allow them to visualize the potential consequences of new regulations and protocols, so they can fine-tune them before enacting them in the real world, effectively avoiding undesirable consequences.

The work in [GGBN13] is supported by an Electronic Institution that handles multiple negotiation protocols in a coherent and flexible way. The Electronic Institution supports contract definitions, modelled as set of formal commitments that can have complex nested structures. The work is complemented with a generic negotiation framework that includes tools to specify system performance indicators, introduce agents in the simulation and allow both human and software agents to participate in the simulations.

According to the authors, a transition from a regulated centralized system to an open virtual market raises the following questions:

1. How to agree on semantic alignments involving multiple ontologies that meet the new requirements of the virtual market?
2. How to recruit agents (or agentified services) to form teams or composed services for the market?
3. How to negotiate in the emerging organization?
4. How the conventions, norms and negotiation protocols of the market change over time, and how participants on these markets react to these changes?
5. How to interpret the outcomes of the market in terms of economic, social and environmental impact, effectively dealing with aspects of the market related to welfare?

And at the same time, requires the following capabilities:

1. Rich ontology and semantics.

2. Entities supporting norm reasoning, monitoring, enforcing and regulation.
3. Organization schemes that are flexible and able to adapt to a changing environment with multiple situations.
4. Coordination, cooperation and dynamic group formation mechanisms
5. Rules for negotiation, argumentation frameworks and conflict resolution techniques.
6. Trust and reputation models and mechanisms.
7. Control and security procedures.
8. A transparent and seamless way to ingrate all the components.

In order to tackle this issues, the main objectives in $m$Water are:

1. Help in finding the conditions and taking the best decisions when designing a water market, effectively providing an environment for testing these parameters.
2. Contribute to deploying a virtual market, to simulate the interplay between intelligent agents, rule enforcing mechanisms and performance indicators. It is possible to test new mechanisms for trust, negotiation, cooperation and argumentation, effectively assessing their impact on market performance indicators.
3. Evaluate the effects of norms in the market, including government norms, local norms and social norms. The idea is helping in developing the appropriate water laws to regulate user actions, effectively promoting the exchange of water resources in the scope of the market.

$m$Water is inspired by the $MAELIA$ [BMJ09] and $NEGOWAT$ [Dur04] projects. Its conceptual model of a virtual market includes scenes (connected among them in networks known as performative structures) and regulations based on structural norms. The model includes the following performative structures:

1. Top performative structure of the market. Including common concepts such as *entitlement* (allowing to trade water rights), *accreditation* (allowing to enter the market and participate in it), *agreement validation* (checking if the agreement complies with the conventions and therefore is valid), *contract enactment* (effectively creating a water transfer contract signed by the actors involved once a valid agreement has been reached) and *annulment* (to deal with a temporary or permanent withdrawal of rights).
2. Trading hall. Where actors become aware of the trading activity, initiate and receive trading proposals, start formal complaint procedures and get informed about anomalous situations (*e.g.,* heavy rains and droughts). It also contains a scene to mediate in disputes about water rights.
3. Trading table. Contains multiple trading scenes, each one corresponding to a trading mechanism or negotiation protocol. The scenes support face to face negotiation, Dutch auction protocols, English auction protocols and blind double auction protocols.
4. Grievances. Once an agreement is active and is being executed it is monitored to asses weather it is being executed properly. Otherwise contract repair actions may apply and conflicts solved via Alternative Dispute Resolution mechanisms [Sou08] and Online Dispute Resolution mechanisms [SBLB01].
5. Arbitration. If a formal complaint is presented by any actor an arbitration process starts. The conflicting parties present their allegations to a jury that passes a resolution on the conflict.

$m$Water simulation procedures are implemented based on the JADE agent platform [JAD] offering flexible and open templates to implement different agent behaviours and norms. The system includes an open norm model, supporting norm violation, detection and resolution via *Grievance* performative structures. The system aims at supporting evolving regulations, beyond parametrised protocols and reusable scenes.

## 2.4 Normative Systems and Health Care

Population ageing is becoming a global problem, as older population (aged 60 years or over) is estimated to grow from the current 11% to 22% by 2050 [Pop12]. Moreover, the cost of supporting an elder is greater than the cost of supporting a child in a ratio of five to three [Uni04], most of this cost being caused by higher health expenses. In the coming years this situation (together with other economic factors) will put great pressure on the national healthcare budgets, mainly because therapies for managing chronic diseases (*e.g.,* diabetes, Parkinson, *etc.*) are performed away from the institutional care setting, typically at home. This distributed approach to daily care requires that elders be capable of *autonomously* taking several different medications at different time intervals over extended periods of time. This can easily lead to forgetfulness or confusion when following the prescribed treatment, specially when the patient is suffering multiple pathologies that require a treatment with a drugs cocktail. This gets worsened in elders suffering from a cognitive impairment. Since medication compliance is a critical component in the success of any medical treatment, this becomes an important problem to tackle for the patient's well-being and the efficient use of resources.

In this context, Assistive Technologies (AT) have been able to provide successful solutions on the support of daily healthcare for elder people, mainly focused on the interaction between the patient and electronic devices. However, the distributed approach that such kind of healthcare has to follow in the current socio-economical setting (*e.g.*, people mobility, online available services, shared costs, hetereogeneous knowledge sources, distributed responsabilities, *etc*) requires more complex AT designs that go further than the interaction with a tool and are able to focus on the relationship between the users and their social environment: caretakers, relatives, health professionals.

The increasing dependence on information technologies in health care organisations has increased the interest in security techniques applied to healthcare. Typically, security is concerned with the protection of information from unauthorized access, either while stored or communicated. It is widely accepted that sensitive medical data (ranging from complex genome information to simple medical records) must be dealt with special care regarding security. It is clear that research on secure access to data will be fundamental in ensuring any software component in general and agents in particular may access or update sensitive information. However, in highly regulated scenarios involving several tasks that must be coordinated by a range of actors, both the roles for some of the actors (*e.g.,* who can act as patient's caregiver) and the way they should or may interact with patients (*e.g.,* who can access patient compliance records) are clearly defined and regulated, and this requires security from a higher level point of view [FD00]. Not only sensitive data must be protected, but the activities carried out by the actors should be controlled, and this implies that the rules and protocols regimenting the medical organisation where the agents operate must be represented, understood by the different agents (either human agents or computational processes) and enforced[Nor90]. Including such rules and protocols into the system will not only enhance security but also social acceptance (from patient's point of view) and professional acceptance (from doctor's point of view). Therefore, the need to use information technologies that comply with pre-defined patterns of behaviour (*i.e.* medical regulations and protocols) arises. Electronic specifications of norms are one of the mechanisms being applied to define and enforce acceptable behaviour of electronic distributed systems which should comply with some (typically human) regulations.

In order to meet the information needs of both health care professionals and patients, health care software systems are deployed. On the one hand, the information provided by these systems needs to be available in real-time and without errors, so the software can effectively provide secure and trustworthy recommendations. On the other hand, health care software systems operate in environments characterized by shared and distributed decision making and management, requiring the communication of complex forms of information and the coordination of different healthcare professionals with a wide range of skills and roles. Therefore, since the properties of Multi-Agent Systems (autonomy, reactivity, pro-activity and sociability [JSW98]) match quite precisely the requirements of such complex scenarios, there is a growing interest in the application of agent-based techniques to the medical domain [NM03].

This section presents a short survey on the existing work in the area of Multi-Agent Systems applied to healthcare, with special emphasis on the works focused on representing and tackling the rules and protocols regimenting the medical organisation where the agents operate.

### 2.4.1 CARREL and CARREL+

CARREL [VSCP$^+$03] is a virtual organization formalized using ISLANDER [EDLCS02] and focused on procuring organs and tissues for transplant. The increasing rate of success in both organ and tissue transplant is leading to an increase in the number of requests, effectively overwhelming the human coordinators at hospitals who are responsible for managing the transplant process. Furthermore, the length of time it takes to process a request can even lead to tissue loss, because tissues exceed their *shelf life*. In the case of organs, the relative scarcity of donors has lead to the creation of international coalitions of transplant organisations, which presents two main issues. First, in order to manage requests at an international level, there is a need to coordinate both surgery teams (typically distributed geographically) and organ delivery. Second, the necessity to accommodate (sometimes conflicting) national and international regulations, legislations and protocols governing the exchange of organs. Please notice that such regulations might even change over time, and Electronic Institutions(*EI*) [VS03], along with the norms governing them, are the key to a system that is able to adapt automatically to changes in regulations.

The main idea behind CARREL is supporting policy makers in designing, implementing and verifying tissue and organ allocation policies, so that survival rates can be maximized. Carrell formalizes tissue and organ allocations processes by specifying interactions between agents (*i.e.* hospital, donor and receiver). This is because ISLANDER interprets an agent-based Electronic Institution as a type of *dialogical system* where all the interactions inside the institution are the composition of (several) dialogic activities (*i.e.* message exchanges). The interactions (known as *illocutions*) are structured through agent group meetings known as *scenes* that abide to defined protocols. The second important element of the ISLANDER formalism is the concept of role. Each agent is assigned several roles, and these roles define the scenes the agent can participate in, and the protocols that should be followed.

CARREL+ [TCM$^+$06] proposes enhancing organ selection process by supporting a more deliberative organ availability assessment. CARREL+ aims at coordinating joint deliberation between donor and recipient agents (representing medical professionals responsible for donor and recipient respectively) in an argument-based dialogue that can evaluate the exchanged arguments. The idea is moving from the actual situation, where deciding weather to offer a particular organ is based exclusively on the assessment of experts at the donor site, to a situation where a joint deliberation takes place between donor and recipient representatives. Therefore, an organ that an expert at the donor site would normally discard can still be transplanted, as long as the recipient agent can successfully argue the organ is viable for the recipient. For ensuring compliance with institutional norms and protocols, the CARREL+ system is able to analyse proposed arguments in order to ensure they are relevant (*i.e.* they follow the specified guidelines and protocols). Finally, please note that both CARREL and CARREL+ are decision support systems, they will provide suggestions and human experts will take the final decisions.

### 2.4.2 Automated monitoring of medical protocols

Controlling the correct application of medical protocols is a key issue in health care environments. In [AAB+03] a system for automated real-time monitoring of medical protocols is proposed. The system consists on two main components. First, a domain-independent language for protocol specification, accompanied by a user-friendly specification tool that that allows health care experts to model a medical protocol and translate into the systems protocol specification language. Second, a semi-autonomous system that understands the protocols and supervises their application. Medical services are modelled as agents, and a medical protocol is interpreted as a negotiation process between agents. The system is able to observe the negotiation, effectively warning about forbidden actions and decisions.

The system is applied to health care environments where every staff person plays one or more roles. A role specifies a particular service (*e.g.,* infirmary, surgery, *etc.* ) and a medical protocol specifies possible interactions between the different services in front of a particular pathology. The protocol can suggest or forbid medical decisions depending on the medical history and evolution of the patient. Suggested actions correspond to medical guidelines and forbidden actions to medical protocols.

The system is composed by three main components:

- A user-friendly graphical environment (known as JAFDIS [ABA+98]) to represent a negotiation process (and by extension, a medical protocol) in a mMulti-Agent System. The environment is grounded in the notion of electronic institution.
- Suitable MAS architectures for real time monitoring medical protocols in distributed health care environments. Agent interactions are performed as message exchanges through a communication layer. *Supervisor agents* track such interactions and validate their validity.
- Robust communication interface providing privacy, integrity and authentication to the process of exchanging information between agents.

### 2.4.3 A Multi-Agent Systems approach for monitoring the prescription of restricted use antibiotics

Hospitals have a specified set of *antibiotics for restricted use* which are aggressive, expensive and only recommended for specific pathologies. Typically, the pharmacy department is responsible for checking prescriptions for this kind of antibiotic. The work in [GPGS+03] presents a Multi-Agent System to monitor and help in the revision of medical prescriptions including antibiotics of restricted use. The system assigns an agent to each patient which is responsible of monitoring medical aspects related to patient's prescribed therapy. A centralized pharmacy agent receives monitored data, analyses it and assigns alternative antibiotic treatments. This will effectively automate a process that is currently being carried out by the pharmacy department. The system is specified using the methodology of electronic institutions ISLANDER [EDLCS02].

The system consists in a patient-centric Multi-Agent System application in a hospital environment. The whole hospital can be viewed as a Multi-Agent System, where each patient has an agent attached that interacts with other kind of agents (*i.e.* pharmacists, physicians, laboratories, *etc.* ). Using electronic institutions allows to model the protocols and interactions that take place among the different agents involved in the application. Agent interactions take place inside scenes, which are basically group meetings composed of a set of agents playing different roles and interacting via a communication protocol. The set of roles allowed to participate in the scene are identified, and any agent playing such roles may enter the scene. To specify the communication protocol, the scene is structured as a graph, where the nodes represent the different states of a conversation and arcs are labelled with utterances belonging to the communication language.

Actors participating in the institution are attached to *governor* agents. *Governors* act as proxies hiding the low-level details of interacting with the institution to the agent. They also control the actions of the agent, according to the current state of the institution, effectively filtering actions that do not abide with institutional norms and protocols.

### 2.4.4 Toward a Conceptual Agent-based Framework for Modelling and Simulation of Distributed Healthcare Delivery Systems

In [CM08] Charfeddine and Montreuil present the $AOE^2$ framework for agent-based modelling and simulation in distributed healthcare delivery systems. They propose a framework that integrates (in a model that is both general and coherent) the main concepts to be considered in order to build an agent-based simulator for the particular domain of health care. An overview of the framework is depicted in *Figure* 2.4. The framework is demonstrated using a real field case: The *Québec* regional COPD (Chronic Obstructive Pulmonary Disease) network.

The $AOE^2$ framework is applied to agent-based simulations, which are abstract representations of the reality. They include a model able to reproduce the behaviour of the system by presenting the decision making entities of the studied system as agents. Multi-Agent System simulated environments combine the benefits of agent systems with those of the simulation modelling approach. They provide researches, policy-makers and managers in health care with a tool for asking *'What-if'* questions, effectively testing different scenarios about the implications of their decisions on the care delivery performance.

The main idea behind the $AOE^2$ framework is focusing in high level conceptual issues regarding health care model development process, while offering a guideline for carrying out this process independently of technical choices. According to authors, using this generic framework abstracted from technical issues, helps in accelerating the model development process for agent-based simulation in distributed health care system. Furthermore the framework minimises the risks of missing any critical element, concept or interaction.

The idea of applying a framework to agent-based simulations in the healthcare domain is appealing. The complexity and dynamics of the domain (*e.g.,* the high degree of

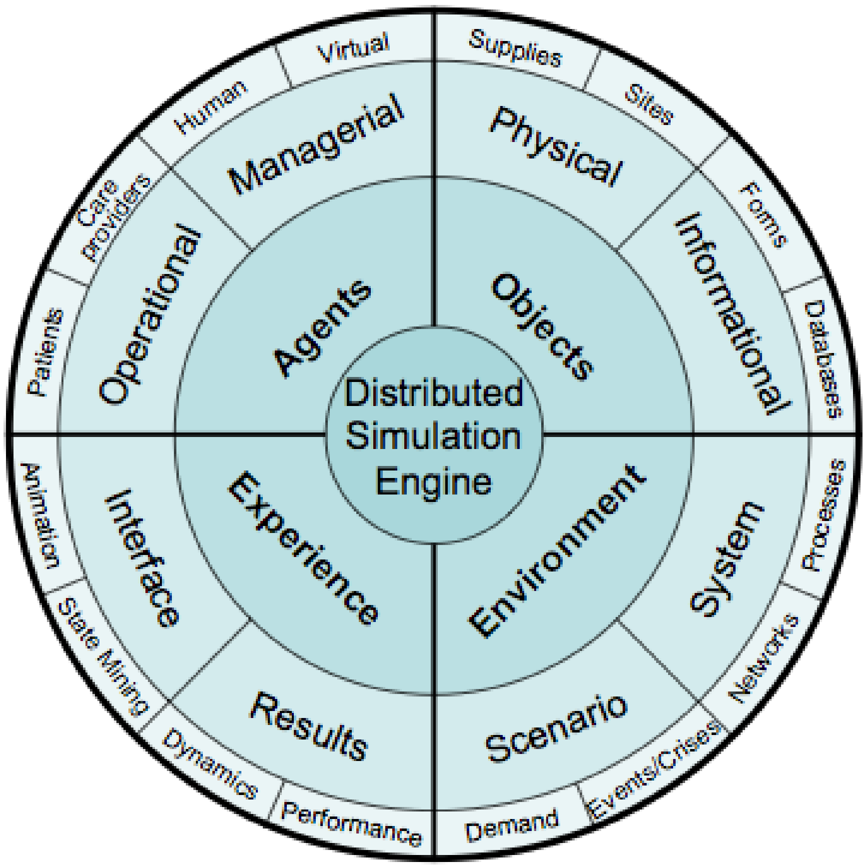



Figure 2.4: Overview of the $AOE^2$ framework

uncertainty inherent to clinical processes, the involvement of multiple distributed service providers and decision makers, *etc.*) make it useful for applying agent-based simulations. Furthermore, the approach is also valid for providing a tool able to asses the possible outcomes of the different actions that can be taken in order to improve the system, making it more efficient or sustainable from an economic point of view.

## 2.5 Discussion

This section will draw conclusions on the different lines of work analysed in this chapter. Two main lines of work for norm monitoring have been analysed in this section, Nir Oren's *et al.* in §2.1.6 and Sergio Alvarez-Napagao *et al.* in §2.1.7, therefore conclusions will focus on these two main lines of work without neglecting the rest of the works analysed.

Nir Oren *et al.* present a theoretical framework, focused on electronic contract monitoring, for determining weather a norm violation takes place. It also supports detection on the occurrences of additional critical states (*e.g.*, norm violation, norm fulfillment, etc.), even allowing to introduce new domain dependent critical states. According to Oren *et al.* the framework can support verification of Normative Systems by forward simulation over the domain and normative environments. Their framework provides a rich language for contract definition and a clear formalisation of the language's semantics.

However, they do not make any reference to the computational cost of their procedure, that seems to be complex, specially if the number of norms and predicates to take into account is considerably high. Indeed, no reference is made on how to efficiently maintain the state of the normative environment. Their framework assumes all agents participating in one contract share the same normative environment. Therefore, their framework can be seen as strict, not designed to cope with inconsistencies among normative environments and not allowing agents to participate in more than one normative environment (that is, one contract) at the same time. For instance, Oren *et al.* framework might have problems dealing with cases where a particular agent has the obligation to perform a given action and, at the same time, will be sanctioned if he performs it, due to the agent being in two inconsistent normative environments at the same time. Thus, their framework might not be suited for scenarios where several local agent theories could coexist, for instance, in non fully observable environments. The framework does not make any reference about what should occur if an obligation is violated, and lacks an outline on how to define the institutional agents that will take care of some core tasks of their framework (such as obligation enforcement, sanctioning, performing repair actions or solving disputes if different local points of view of a particular fact arise).

In general, analysing Oren's approach has been useful in the scope of this proposal, because it provides an easy to understand and well-structured language for defining clauses on the contracts (norms, for the scope of this proposal) as well as meta-information associated to such norms (such as contract participants or the contract state). In our opinion, the most relevant contribution of Oren *et al.* approach to our proposal is the definition of the formal operational semantics concerning the different states on contract's clauses, specially in the scope of our work that will focus mainly on monitoring these states. Putting together a clear norm formalisation like the one performed by Oren with a more expressive norm life-cycle like the one presented by Fabiola López y López in [LL03] seems an interesting line of future work.

Alvarez-Napagao *et al.* present a theoretical framework and a implementation (analysed in §2.1.7) focused on allowing agents to perceive the current normative state of the environment, and allowing the environments to detect norm violations and enforce sanctions. The main idea is the detection of normative states by monitoring past events and checking them against a set of active norms. This is achieved via production systems with a forward-chaining rule engine that automatically trigger the normative state as new events arrive into the system. This allows for fully decoupling the procedure for normative state monitoring from the procedure for agent reasoning, while allowing the agents to interpret institutional facts from brute ones and decide what ought to be done. In short, Alvarez-Napagao *et al.* work allows for combining constitutive norms with regulative ones. They present a clear and bold line of work that can be seen as an improvement of the approach by Oren *et al.* First, it supports both constitutive and declarative norms. Second, it has proved it is as expressive as conditional deontic statements with dead-lines. Third, and most important, reducing the framework to production rules on production systems allows for guaranteeing the computational cost is constant in the best case and linear to the number of productions contained in the rules in the worst case. It might not be efficient enough on scenarios with a heavy load of events (or rules) but formally stating a computational cost is a step ahead regarding Oren's work. Finally, Alvarez-Napagao *et al.* proposal contains a proof of concept prototype that facilitates the task of testing his framework on real world scenarios.

However, Alvarez-Napagao *et al.* approach presents a feature that can become problematic on some scenarios. It needs to perform a reduction process from the institutional specification to the production rules he uses in his production system. Alvarez-Napagao *et al.* do not make any reference to the computational cost of this procedure, but regarding this fact, he has two factors on his favour. First, the process does not seem to be complex, so one can assume it can be run efficiently enough. Second, as most scenarios have static institutional specifications, it should be performed only once in most scenarios. However, having to run this reduction process can be a problem on scenarios with dynamic institutional definitions or in multi-institutional scenarios, where the monitoring process has to jump from one institutional definition to another. This is mainly due the fact Alvarez-Napagao's is not able to perform changes to the institutional definition on the fly, without having to stop the monitoring process. In fact, supporting this dynamic normative changes seem like an interesting line of work. Another possible improvement lies on providing efficient access to the log of past events. As Alvarez-Napagao's approach requires keeping a memory of past events, accessing the memory can be costly in case the scenario contains a heavy load of events or in case the monitoring system has been running for a long time. Implementing efficient access to the event log and advanced procedures for discarding past events seem like an interesting line of work for improving Alvarez-Napagao's approach. Just like in the case of Oren's framework, looks like Alvarez-Napagao's approach could benefit from more expressive normative states too, improving the norm life-cycle he defines with richer state definitions, like the ones presented by Fabiola López y López in her proposal [LL03].

In general, we consider that analysing Alvarez-Napagao's approach has been useful in the scope of this proposal, because it provides a fully supported language for monitoring the state of a normative environment. The language covers all the aspects, from formalization and reduction to production system rules to implementation. Yet it contains room for improvement. It provides a prototype that can be adapted and improved providing a effective continuation to Alvarez-Napagao's line of work. All in all, we have decided to use it as a starting point (and even a base system) to be used during the development of my line of research for my PhD Thesis.

**Opera** and **Opera**+ present a framework and methodology for defining both the social structure and the normative structure of an organization. Therefore, our proposal can benefit from **Opera** and **Opera**+ when using formal models of organizations (including concepts such as roles and objectives) and norms. However, neither **Opera** nor **Opera**+ provide details about the structure or architecture of the normative framework. Furthermore, they do not seem to support dynamic Normative Systems.

Aucher *et al.* framework for supporting dynamic logic contexts (analysed in §2.2.2) provides an interesting analysis on the modifications to be performed to logic contexts in order to support norm promulgation and derogation. Its strongest point is the way it is modelling obligations, although we miss the same analysis regarding permissions (or prohibitions), which would have clarified how to fully express deontic statements. We consider providing an expansion to the model with support for such elements would be an interesting research work. Another possible improvement would be allowing to differenciate between *ex tunc* and *ex nunc* promulgation and derogation, allowing for a fully expressive normative context expansion and contraction.

The framework formalizing changes in legal systems proposed by Governatori *et al.* (analysed in §2.2.1) provides an interesting analysis on the alternatives available for supporting two normative context change operations. The analysis is complete, as it explores many formal alternatives, effectively summarising the possible problems one might encounter when trying to formalize such changes. However, we consider a refinement of the framework is required, mainly because:

- We have doubts about the framework's applicability at a practical level. Leaving old versions of norm repositories allows for effectively modelling abrogation. However, we consider this approach might have a negative impact regarding performance, specially if many abrogation operations have taken place and the number of versions (norm repositories) is high.
- The framework focuses on normative context contraction (removing norms) and does not take into account normative context expansion (adding norms). It means support for norm update (where norm update is understood as removing the old version of the norm and adding the new version) is not supported.

During this chapter we have also analysed Normative Systems applied to environmental management scenarios in §2.3 and to health care scenarios in §2.4. On the one hand, theses works are not related to the framework we propose, but to the application of our framework to the testing scenarios we propose. On the other hand, these related works are domain specific (applied to environmental management and healthcare). Therefore, we will put these related works in contrast with our proposal in sections §4.7 and §5.6 for environmental management scenarios and health care scenarios respectively. For simplicity, we also summarise the conclusions drawn from this comparison in §6.3.

The next chapter will provide a formal definition of our framework for computing normative states. The framework effectively allows for performing expansion and contraction operations on a normative context at runtime, that is, without having to stop the monitoring process. The formal definition is provided by introducing a base framework for normative monitoring and then extending it with support for dynamic normative contexts.

CHAPTER 3

# Building Support for Dynamic monitoring systems

## 3.1 Introduction

In the last 15 years, most of the research on normative environments has focused on norm specifications that are static and stable, and do not change over time. Norms are designed off-line, when the system to be governed is not running, and norms do not change at run time. This approach can be analysed both from agent perspective and from institutional perspective.

On the one hand, this approach is not appropriate from an agent perspective. During their life time, agents may enter and leave several institutional contexts, each with its different normative framework. Furthermore, agents may even be operating in several institutional contexts simultaneously, or in contexts where more than one normative specification applies. For instance, an agent under the Catalan institutional context is affected by the Spanish and the European institutional contexts as well. Therefore, mechanisms where normative specifications can be added to the agents knowledge base at run time are required. Then, such specifications can be integrated in the agent's reasoning cycle, allowing it to interpret institutional facts from brute ones, and to decide what ought to be done.

On the other hand, this approach is also not appropriate from a institutional perspective. Most electronic institutions are designed off-line and will maintain a monolithic set of norms, regimenting them. However, electronic institutions are one of the mechanisms being applied to define and enforce acceptable behaviour of distributed electronic systems which should comply with some human regulations. As in human legal systems, it is easy to foresee that such electronic normative environments will not be static, but will have to evolve through time as regulations change to adapt to new situations and behaviours. Therefore, the need for electronic institutions able to evolve the set of norms, regimenting them on-line arises.

Our proposal consists on extending an existing base framework with operations for the expansion and contraction of the normative context, and adapting our architecture to support such operations. We aim to achieve real-time expansion and contraction of the normative context, without having to stop monitoring the state of the world and inferring information about it.

This chapter is structured as follows. We start by presenting an example scenario. Later, we present the base formal framework we will extend during this chapter. Then we formalise a framework for norm change, providing a formal definition of the operations to be supported. We do this by extending the base framework with operations that allows us to perform normative context modifications at run-time. The chapter goes on by providing the formal algorithms for supporting expansion and contraction operations. Then, the chapter provides an architectural design that includes the definition of the different components in the architecture including interfaces for norm update, monitoring components (with special emphasis on the monitor's knowledge base) and institutional agents. Later, the chapter instantiates the formal algorithms for supporting expansion and contraction operations, providing implementations details. The chapter also provides an instantiation of the architecture, with particular technological solutions selected for the different components. Finally, conclusions are drawn. Finally we present a discussion on the framework presented in this chapter relating it to other relevant lines of work analysed as state of the art. We also draw some conclusions.

## 3.2 Example Scenario

This section defines a sample scenario we will use for providing some examples along the chapter. We illustrate our approach via a scenario that models a simplified version of the 2005 Spanish smoking law that has been recently amended. Basically, the 2005 law obliges bars and restaurants with a size bigger than $100m^2$ to provide an isolated area for smoking customers. They will incur in a violation if they do not fulfil this obligation, the violation is considered as repaired once the bar abilitates an area for their smoking customers. The amended 2011 law forbids bars and restaurants to have any smoking area. They will incur in a violation if they have a smoking area, and the violation will be considered as repaired once the bar removes the smoking area. In the scenario presented we define the set of bars as $[B_1, \dots B_k] \in \mathcal{B}$.

Once we have depicted the example we are ready to present the formal framework we will use as basis.

## 3.3 Basic Formal Normative Model

Keeping track of the normative state in a context can be a cumbersome task. Some works such as [ÁNAVSD10] (analysed in §2.1.7) propose a reduction from deontic norms to general production systems. A representation based on production systems is easier to use at run-time. This allows building a norm monitoring mechanism that can be used both by agents to perceive the actual normative state of the environment and by institutions to detect norm violations and enforce sanctions. The production system can be configured at run-time by adding both abstract organisational specifications (*i.e.*, regulative norms) and sets of counts-as rules (*i.e.*, constitutive norms). In such works, the detection of normative states is a passive procedure that consists in monitoring past events and checking them against a set of active norms.

A clear advantage of a norm monitoring mechanism implemented using general production systems is that the efficiency of the system is bound to the complexity of the system. That is, linear to the number of productions contained in the rules in the worst case and constant in the best case.

In this section we present the formalism for monitoring Normative Systems which we will use in the rest of the chapter. Our formalism is an extension of the one by Alvarez-Napagao introduced in §2.1.7[1] and analysed in detail in §A.2.

We assume the use of a predicate based propositional logic language $\mathcal{L}_O$ with predicates and constants taken from an ontology $O$, and the logical connectives $\{\neg, \vee, \wedge\}$. The set of all possible well-formed formulæ of $\mathcal{L}_O$ is denoted as $wff(\mathcal{L}_O)$ and we assume that each formulæ from $wff(\mathcal{L}_O)$ is normalised in Disjunctive Normal Form (*DNF*). Formulæ in $wff(\mathcal{L}_O)$ can be partially grounded, if they use at least one free variable, or fully grounded if they use no free variables.

We define the *state of the world* $s_t$ as the set of predicates holding at a specific timestamp $t$, where $s_t \subseteq O$, and we denote $\mathcal{S}$ as the set of all possible states of the world, where $\mathcal{S} = \mathcal{P}(O)$. We call *expansion* $F(s)$ of a state of the world $s$ as the minimal subset of $wff(\mathcal{L}_O)$ that uses the predicates in $s$ in combination of the logical connectives $\{\neg, \vee, \wedge\}$. We define a substitution instance $\Theta = \{x_1 \leftarrow t_1, x_2 \leftarrow t_2, ..., x_i \leftarrow t_i\}$ as the substitution of the terms $t_1, t_2, ..., t_i$ for variables $x_1, x_2, ..., x_i$ in a formulæ $f \in wff(\mathcal{L}_O)$. Thus, $\Theta(f(x_1, x_2, ..., x_i)) \equiv f(t_1, t_2, ..., t_i)$. We denote as $\vartheta_{(wff(\mathcal{L}_O),\mathcal{S})}$ the set of all possible substitution instances containing the variables in $wff(\mathcal{L}_O)$ and the terms in $\mathcal{S}$.

**Definition 2** (Regulative Norm) *A 'regulative norm' $n$ is a tuple $n = \langle f_A, f_M, f_D, f_w, w\rangle$, where*

- $f_A, f_M, f_D, f_w \in wff(\mathcal{L}_O)$, $w \in O$,
- $f_A, f_M, f_D$ *respectively represent the activation, maintenance, and deactivation conditions of the norm.*
- $f_w$ *is the explicit representation of the target of the norm, and $w$ is the subject of the norm (role or agent).*

□

A regulative norm is defined in an abstract manner, affecting all possible participants enacting a given role. Whenever a regulative norm is active, we will say that there is a *norm instance* $ni = \langle n, \theta\rangle$ for a particular regulative norm $n$ and a substitution instance $\Theta$.

We can formalise the norms of Definition 2 as the equivalent deontic expression (using the formalism of the inference rules[DBDM04]):

**Property 1** *A regulative norm is considered fulfilled if, and only if:*

$$f_A \rightarrow [O_w(E_w f_w \leq \neg f_M) \,\mathcal{U}\, f_D]$$

[1]The main differences between our formalisation and the one in §2.1.7 are the inclussion of constitutive norms as first-level elements in our framework, and the addition of institutional powers (including normative powers and constitutive powers).

*Norm* $N_0$: Let $B_i \in \mathcal{B}$ be a Bar, with a size $\geq 100m^2$. Since Jan 2005 the bar has one month to set up a smoking area
*Sanction* $S_0$: The bar must remain closed until a smoking area is set up.

| | |
|---|---|
| *Activation Condition* $N_0$ | $hasSize(B_i) \geq 100m^2$ |
| *Expiration Condition* $N_0$ | $isTime(Feb\, 2005)$ |
| *Maintenance Condition* $N_0$ | $True$ |
| *Deadline* $N_0$ | $hasSmokingArea(B_i)$ |
| *Activation Condition* $S_0$ | $isViolated(N_0, B_i)$ |
| *Expiration Condition* $S_0$ | $hasSmokingArea(B_i)$ |
| *Maintenance Condition* $S_0$ | $isClosed(B_i)$ |
| *Deadline* $S_0$ | $True$ |

Figure 3.1: Example of regulative norm specification

*where* $E_a p$ *means that agent* $a$ sees to it that (stit) $p$ *becomes true and U is the CTL* until operator.*

Intuitively, Property 1 states that after the activation of a regulative norm, the subject is obliged to see to it that the target becomes true before the maintenance condition is negated (either the deadline is reached or some other condition is broken) until the norm is deactivated (which is either when the norm is fulfilled or has otherwise expired). Intuitively, the unfulfillment of the obligation of a regulative norm $N_i$ entails the activation of a sanction, known as violating handling norm $S_i$. Formally:

**Definition 3** (Violation handling norm) *A norm* $n' = \langle f'_A, f'_M, f'_D, f'_w, w' \rangle$ *is a violation handling norm of a regulative norm* $n = \langle f_A, f_M, f_D, f_w, w \rangle$*, denoted as* $n \rightsquigarrow n'$ *iff* $f_A \wedge \neg(f_M \mathcal{U} f_D) \vdash f'_A$ $\square$

Violation handling norms are special in the sense that they are only activated once a regulative norm is violated. They are used as *sanctioning norms*, if they are to be fulfilled by the norm violating actor (*e.g.*, the obligation to pay a fine if the driver broke a traffic sign), or as *reparation norms*, if they are to be fulfilled by an institutional actor (*e.g.*, the obligation of the authorities to fix the broken traffic sign).

*Figure* 3.1 shows an example of a regulative norm specification following our formalism. The norm is introduced in the system in January 2005. The norm activates for bars with size $\geq 100m^2$. The norm expires a month after entering the system (*i.e.* February 2005). There is no maintenance condition in the norm, therefore, setting it to $True$ ensures the condition is always fulfilled. The deadline (*i.e.* the state of the world to be accomplished before the expiration condition) states the bar has set up a smoking area. The sanction $S_0$ is an example of a violation handling norm. The sanction activates when an instance of the regulative norm $N_0$ is violated. The sanction expires once the bar has a smoking area. And the maintenance condition states the bar must remain closed while the sanction is active, effectively promoting a behaviour where the smoking area is set up. There is no deadline condition in the norm, therefore, setting it to $True$ ensures the condition is always fulfilled. Please, notice that we could define a more stringent sanction $S'_0$ for bars violating $S_0$ (*e.g.*, Event processed:

$$\frac{e_i = \langle \alpha, t, p \rangle}{\langle s \rangle \overset{ep}{\rhd} \langle s \cup \{p\} \rangle} \tag{3.1}$$

Constitutive norm activation:

$$\frac{\begin{array}{c}\Theta(\gamma_1) \equiv f \qquad \Theta(\gamma_2) \notin s \\ \langle \gamma_1, \gamma_2, s_i \rangle \in C \qquad f \in F(s) \qquad s_i \subseteq s\end{array}}{\langle s \rangle \rhd \langle s \cup \{\Theta(\gamma_2)\} \rangle} \tag{3.2}$$

Constitutive norm deactivation:

$$\frac{\begin{array}{c}\Theta(\gamma_1) \equiv f \qquad \Theta(\gamma_2) \in s \\ \langle \gamma_1, \gamma_2, s_i \rangle \in C \qquad f \in F(s) \qquad s_i \nsubseteq s\end{array}}{\langle s \rangle \rhd \langle s - \{\Theta(\gamma_2)\} \rangle} \tag{3.3}$$

Regulative norm instantiation:

$$\frac{activated(n, \Theta) \qquad n \in N \qquad \neg \exists n' \in N, n' \rightsquigarrow n \qquad \langle n, \Theta \rangle \notin is}{\langle is \rangle \overset{nii}{\rhd} \langle is \cup \{\langle n, \Theta \rangle\} \rangle} \tag{3.4}$$

Norm instance violation:

$$\frac{\neg maintained(\langle n, \Theta \rangle) \qquad NR = \bigcup_{n \rightsquigarrow n'} \langle n', \Theta \rangle \qquad n \in N \qquad \langle n, \Theta \rangle \in is \qquad \langle n, \Theta \rangle \notin vs}{\langle is, vs \rangle \overset{niv}{\rhd} \langle (is - \{\langle n, \Theta \rangle\}) \cup NR, vs \cup \{\langle n, \Theta \rangle\} \rangle} \tag{3.5}$$

Norm instance fulfilled:

$$\frac{deactivated(n, \Theta') \qquad n \in N \qquad \langle n, \Theta \rangle \in is \qquad \Theta' \subseteq \Theta}{\langle is, fs \rangle \overset{nif}{\rhd} \langle is - \{\langle n, \Theta \rangle\}, fs \cup \langle n, \Theta \rangle \rangle} \tag{3.6}$$

Norm instance violation repaired:

$$\frac{\langle n', \Theta \rangle \in fs \qquad n, n' \in N \qquad n \rightsquigarrow n' \qquad \langle n, \Theta \rangle \in vs}{\langle vs, rs \rangle \overset{nir}{\rhd} \langle vs - \{\langle n, \Theta \rangle\}, rs \cup \{\langle \langle n, \Theta \rangle, \langle n', \Theta \rangle \rangle\} \rangle} \tag{3.7}$$

Figure 3.2: Inference rules for the transition relation $\rhd$

bars opening while the sanction $S_0$ is active may have their license revoked and be economically sanctioned). Then we can define another sanction for bars violating $S_0'$ (*e.g.*, $S_0''$ for bars opening without a license and $S_0'''$ for bars not paying economic sactions) and so on. Our formalism allows to effectively define complex and complete sets of behaviour enforcing policies. Chapters §4 and §5 contain more norm specification examples based on our use cases.

One common problem for the monitoring of normative states is the need for an interpretation of brute events as institutional facts, also called constitution of social reality[Gro07]. The use of constitutive norms helps solving this problem. Constitutive norms (also known as constitutive rules) are contextual rules mapping uninterpreted brute facts into interpreted institutional facts.

In the scope of agents literature, constitutive norms are typically formalized in sets of counts-as rules [Gro07]. Counts-as rules are multi-modal statements of the form $[c](\gamma_1 \rightarrow$

$\gamma_2$), read as "in context $c$, $\gamma_1$ *counts-as* $\gamma_2$". In our proposal, we consider a context as a set of predicates, that is, as a possible subset of a state of the world:

**Definition 4** (Constitutive norm) *A constitutive norm is a tuple* $c = \langle \gamma_1, \gamma_2, s \rangle$, *where* $\gamma_1, \gamma_2 \in wff(\mathcal{L}_O)$, *and* $s \subseteq O$.
*We also define constitutive norms in function form, formally:*

$$counts\_as(\gamma_1, \gamma_2, s)$$

*Where we can omit the context if it is obvious* (e.g., *the institution contemplates only one context).*

$$counts\_as(\gamma_1, \gamma_2)$$

*Both forms are equivalent:*

$$\langle \gamma_1, \gamma_2, s \rangle \stackrel{\text{def}}{=} counts\_as(\gamma_1, \gamma_2, s)$$

□

It is important to note here that, while regulative norms may be instantiated several times, constitutive norms have no instantiations. Constitutive norms may be added (becoming active) and removed (becoming inactive). It is also worth to remark that while regulative norms can be violated (that is why we define violating handling norms) constitutive norms cannot.

The set of constitutive norms that are active in a particular context in denoted as $C$. Although the definition of counts-as in [Gro07] assumes that both $\gamma_1$ and $\gamma_2$ can be any possible formulæ in our work we limit $\gamma_2$ to a conjunction of predicates. This will ensure every well-formed formulæ is on a standard *Disjunctive Normal Form*.

**Definition 5** (Institution) *Following the definitions above, we define an* institution *as a tuple of regulative norms, roles, participants, constitutive norms, and an ontology:*
$I = \langle N, R, P, C, O \rangle$
*where:*

$N$ *is a set of norms*

$R$ *is a set of roles*

$P$ *is a set of participants*

$C$ *is a set of counts-as rules*

$O$ *is a set an Ontology*

□

In order to track the normative state of an institution at any given point of time, the state of each of the norms inside the context should be tracked. In order to ease this task, we define the following sets:

**Definition 6** (Normative State)

*$IS$ is an instantiation set*

*$FS$ is a fulfilment set*

*$VS$ is a violation set*

*$RS$ is a repairment set*

*Each of these sets contains norm instances $\{\langle n_i, \Theta_j \rangle, ..., \langle n_{i'}, \Theta_{j'} \rangle\}$*

*The normative state is a tuple formed by these sets $NS = \{\langle IS, FS, VS, RS \rangle\}$*

□

*Definition* 6 defines the possible normative states. In order to operationally define the states a norm instance may be in, we adapt the semantics for normative states from [OPVS+09] as follows:

**Definition 7** (Regulative Norm Lifecycle) *Let $ni = \langle n, \Theta \rangle$ be a regulative norm instance, such that $n = \langle f_A, f_M, f_D, w \rangle$, and $s$ be a state of the world with an expansion $F(s)$. Then we define the lifecycle for a norm instance $ni$ by the following normative state predicates:*

$activated(ni) \Leftrightarrow \exists f \in F(s), \Theta(f_A) \equiv f$
$maintained(ni) \Leftrightarrow \exists \Theta', \exists f \in F(s), \Theta'(f_M) \equiv f \wedge \Theta' \subseteq \Theta$
$deactivated(ni) \Leftrightarrow \exists \Theta', \exists f \in F(s), \Theta'(f_D) \equiv f \wedge \Theta' \subseteq \Theta$
$instantiated(ni) \Leftrightarrow ni \in IS$
$violated(ni) \Leftrightarrow ni \in VS$
$fulfilled(ni) \Leftrightarrow ni \in FS$
$repaired(ni, ni') \Leftrightarrow \langle ni, ni' \rangle \in RS$

*Where $IS$ is the instantiation set, $FS$ is the fulfillment set, $VS$ is the violation set, and $RS$ is the set of those norm instances $ni'$ that have repaired a norm instance $ni$.* □

**Definition 8** (Constitutive Norm Lifecycle) *Let:*

*$c = \langle \gamma_1, \gamma_2, s \rangle$ be a constitutive norm.*
*$I = \langle N, R, P, C, O \rangle$ be an institution.*

*We define two possible state predicates for the constitutive norm:*

$$active(c) \Leftrightarrow c \in C$$
$$inactive(c) \Leftrightarrow c \notin C$$

□

**Definition 9** (Event) *An event $e$ is a tuple $e = \langle \alpha, t, p \rangle$, where*

- *$\alpha \in O$, an actor of the system,*
- *$t$ is the timestamp of the reception of the event, and*
- *given a fully grounded subset of the set of states of the world $p' \in S : p = p' \vee p = \neg p'$*

$\square$

We define $E$ as the set of all possible events, $E = \mathcal{P}(P \times S)$. We define $H$ as the subset of $E$ that has been observed by the monitor, that is, a history.

From these definition we can formalise the concept of *Normative Monitor* and the concept of *Labelled Transition System for a Normative Monitor* as follows:

**Definition 10** (Normative Monitor) *A Normative Monitor $M_N$ for a set of norms $N$ is a tuple $M_N = \langle N, S, IS, VS, FS, H \rangle$. Where:*

- *$S = \mathcal{P}(O)$.*
- *$IS = \mathcal{P}(N \times S \times Dom(S))$.*
- *$VS = \mathcal{P}(N \times S \times Dom(S))$.*
- *$FS = \mathcal{P}(N \times S \times Dom(S))$.*
- *$H \in E$ is the history of events that has been observed by the monitor as defined before.*

$\square$

$\Gamma_{M_N}$ is the set of all possible configurations of a Normative Monitor $M_N$.

**Definition 11** (Labelled Transition System) *The* Labelled Transition System $LTS_{M_N}$ *for a Normative Monitor $M_N$ is defined by $LTS_{M_N} = \langle \Gamma_{M_N}, L, \rhd \rangle$ where*

- *$L = \{ep, nii, niv, nif, nir\}$ is a set of labels, respectively representing* event processed*,* regulative norm instantiation*,* norm instance violation*,* norm instance fulfilled*, and* norm instance violation repaired*, and*
- *$\rhd$ is a* transition relation *such that $\rhd \subseteq \Gamma_{M_N} \times L \times \Gamma_{M_N}$*

$\square$

The inference rules for the transition relation $\rhd$ are described in Figure 3.2.

This formalism, as shown in §2.1.7, has been reduced to the semantics of general production systems and an implementation in DROOLS is already available.

As seen in *Definition*2 our approach models regulative norms as tuples including activation, maintenance and deactivation conditions. These norms also include a target (also known as deadline condition) and a subject, that is, the role or particular agent responsible of seeing to it that the regulative norm is complied with. In this aspect, we support two types of obligations.

First, the obligation to achieve, at least once, a state of the world. The obligation is active only after the activation condition holds true. The obligation must be fulfilled before the deactivation condition holds true. The typical example is 'a citizen has the obligation to pay his taxes at least once each year. Taxes must be paid once the fiscal year opens, not before. Taxes must be paid before the fiscal year ends, not after'. As seen in *Figure* 3.3 we use the target of the norm (*i.e.* deadline condition) to model a state of the world to be achieved at least once since the activation of the regulative norm and before the

*Norm* $N_0$: Let $citizen$ be a particular citizen, and $taxes(citizen)$ the taxes a citizen has to pay. We assume the fiscal year starts on 01-Jan and ends on 31-Dec

| | |
|---|---|
| *Activation Condition* $N_0$ | $isDate(01-Jan)$ |
| *Expiration Condition* $N_0$ | $isDate(31-Dec)$ |
| *Maintenance Condition* $N_0$ | $True$ |
| *Deadline* $N_0$ | $paid(citizen, taxes(citizen))$ |

Figure 3.3: Example of obligation with deadline

*Norm* $N_0$: Let $citizen$ be a particular citizen, $a_1$ and $a_2$ artefacts and $using(x, y)$ a function returning true if a person $x$ is using an artefact $y$ and false otherwise. We can model via constitutive rules which artefacts count as a proper helmet and a as a motorbike. We assume we are in the institutional context X

| | |
|---|---|
| *Activation Condition* $N_0$ | $using(citizen, a_1) \wedge countsAs(a_1, motorbike, X)$ |
| *Expiration Condition* $N_0$ | $\neg using(citizen, a_1)$ |
| *Maintenance Condition* $N_0$ | $using(citizen, a_2) \wedge countsAs(a_2, helmet, X)$ |
| *Deadline* $N_0$ | $True$ |

Figure 3.4: Example of obligation with maintenance condition

deactivation of the regulative norm. As there is no maintenance condition we set it to $True$ to reflect it is always fulfilled.

Second, the obligation to maintain a state of the world while the norm is active. The obligation is active only after the activation holds true. The obligation holds until the deactivation condition becomes true. The typical example is 'a citizen riding a motorbike has the obligation to use a helmet. The obligation holds once the citizen is on a motorbike, not before. The obligation holds while the citizen is on a motorbike, once he abandons it, the obligation to wear a helmet does not hold'. As seen in *Figure* 3.4 we use the maintenance condition to model a state of the world to be achieved at least once since the activation of the regulative norm and before the deactivation of the regulative norm. As there is no deadline condition we set it to $True$ to reflect it is always fulfilled.

Please note that it is convenient to distinguish both types of obligations, because they are very different from a functional perspective. It makes no sense to force citizens to continuously pay taxes during the fiscal year and it makes no sense to encourage motorbike riders to wear a helmet just during a brief period of time while driving a motorbike. Also, please notice how both types of obligations can be easily combined in the same norm.

We can specify the condition to be met in order to violate a norm. Intuitively, our regulative norms are violated when the following conditions are met:

1. The activation condition holds true. If the regulative norm is not active, it can not be violated.
2. Condition 1 holds and the expiration condition holds true before the deadline condition holds true. As seen in *Definition* 9 the events in our system have the specific

representation of the time performing it when the events happen. Informally, the expiration condition holds true before the deadline condition when the event registering the expiration condition have a smaller timestamp than the event registering the deadline condition.

3. The activation condition evaluates true and the expiration condition false and the maintenance condition false. While the regulative norm applies, the maintenance condition has been violated.

Formally:

**Definition 12** ***(Obligation Violation)***

$$
\begin{gathered}
\textit{Given an obligation in the form of regulative norm } n = \langle f_A, f_M, f_D, f_w, w\rangle \\
\textit{Given an event of the form } e = \langle \alpha, t, p\rangle \\
\textit{Given the set of times as } [\tau_1, \ldots \tau_k] \in \mathrm{T} \\
\textit{Given the free variable \_ which matches any value.} \\
Violated(n = \langle f_A, f_M, f_D, f_w, w\rangle) \textit{ iff} \\
\langle \_, \tau_1, f_A\rangle \wedge \\
\Big( \big( (\neg \exists \tau_2 \in \mathrm{T} : \langle \_, \tau_2, f_D\rangle \wedge T2 \geq T1) \wedge \\
(\exists \tau_3 \in \mathrm{T} : \langle w, \tau_3, \neg f_M\rangle \wedge T3 \geq T1) \big) \\
\vee \\
\big( (\exists \tau_2 \in \mathrm{T} : \langle \_, \tau_2, f_D\rangle \wedge T2 \geq T1) \wedge \\
(\neg \exists \tau_3 \in \mathrm{T} : \langle w, \tau_3, f_w\rangle \wedge T3 \geq T1) \big) \Big)
\end{gathered}
$$

□

In our proposal permissions are modelled as the dual operator of the prohibition. We allow norm designers to model prohibitions (assuming states of the world not explicitly prohibited are permitted) or model permissions (assuming states of the world not explicitly prohibited are permitted). Therefore, if a particular state of the world $\varphi$ is permitted $\neg\varphi$ is prohibited. Analogously if $\varphi$ is prohibited $\neg\varphi$ is permitted. Formally:

**Definition 13** (Permission and prohibition duality) *Let $wff(LO)$ be the set of all well-formed formulæ. $\forall \varphi \in wff(LO)$ $P(\varphi)$ stands for $\varphi$ is permitted in our normative context. Analogously $F(\varphi)$ means $\varphi$ is prohibited (*i.e.* forbidden) in our normative context.*

*Let $wff(LO)_P \subseteq wff(LO)$ be the set of permitted states of the world, that is $\forall \varphi \in wff(LO)_P : P(\varphi)$. Analogously let $wff(LO)_F \subseteq wff(LO)$ be the set of prohibited states of the world, that is $\forall \varphi \in wff(LO)_F : F(\varphi)$. As we know everything not permitted is prohibited and vice versa we know $wff(LO)_F \cap wff(LO)_P = \emptyset$ and $wff(LO)_F \cup wff(LO)_P = wff(LO)$.*

*Therefore, given the set of permitted states of the world $wff(LO)_F$ we can define the set of prohibited states of the world $wff(LO)_P$ and vice versa. It allows us to express prohibitions as permissions and permissions as prohibitions. Formally, given a set of states of the world $wff(LO)_S \subset wff(LO)$:*

$$P(S) \stackrel{\text{def}}{=} F(S') \wedge S' = wff(LO) - S$$

$$F(S) \stackrel{\text{def}}{=} P(S') \wedge S' = wff(LO) - S$$

□

As *Definition* 13 demonstrates, permissions in our proposal can be effectively modelled as prohibitions, and prohibitions as permissions.

*Norm* $N_0$: Let $citizen$ be a particular citizen, $a_1$ and $a_2$ artefacts and $using(x, y)$ a function returning true if a person $x$ is using an artefact $y$ and false otherwise. We can model via constitutive rules which artefacts count as headphones and as a motorbike. We assume we are in the institutional context X

| | |
|---|---|
| *Activation Condition* $N_0$ | $using(citizen, a_1) \wedge countsAs(a_1, motorbike, X)$ |
| *Expiration Condition* $N_0$ | $\neg using(citizen, a_1)$ |
| *Maintenance Condition* $N_0$ | $\neg(using(citizen, a_2) \wedge countsAs(a_2, headphones, X))$ |
| *Deadline* $N_0$ | $True$ |

Figure 3.5: Example of prohibition using maintenance condition

In our proposal, prohibitions can be seen as states of the world to be avoided while the regulative norm applies (*i.e.* between activation and deactivation condition). Intuitively they can be expressed as the obligation to maintain a state of the world while the norm is active, where this state of the world is the negation of the prohibition. The typical example is *'a citizen riding a motorbike has the prohibition to use headphones'*. The prohibition holds once the citizen is on a motorbike, not before. The prohibition holds while the citizen is on a motorbike, once he abandons it, the prohibition to use headphones does not hold'. As seen in *Figure* 3.5 we use the maintenance condition to model the negation of the state of the world to avoided. As there is no deadline condition we set it to $True$ to reflect it is always fulfilled.

Please notice that we model prohibitions as obligations. The model is more simple, because we are never using the deadline condition (we only model type 2 obligations). Formally:

**Definition 14** (Modelling prohibitions as obligations)

*According to Definition2 a norm in our model is is a tuple* $n = \langle f_A, f_M, f_D, f_w, w\rangle$ *with activation, maintenance and deactivation conditions, as well as target (deadline) and subject.*

*According to Definition9 an event in our model is is a tuple* $e = \langle \alpha, t, p\rangle$*, with the actor responsible of the event, a timestamp and the state of the world representing the event.*

*Assuming we want to forbid* $\varphi$ *for agent* $w$ *while the norm is active (*$f_A$ *holds and* $f_D$ *does not hold) our norm will be of the form* $n = \langle f_A, \neg\varphi, f_D, f_w, w\rangle$*.*

*Formally prohibition violation in our model is defined as follows:*

$$\begin{gathered}\textit{Given the set of times as } [\tau_1, \ldots \tau_k] \in \mathrm{T} \\ \textit{Given the free variable \_ which matches any value.} \\ Violated(n = \langle f_A, \neg\varphi, f_D, f_w, w\rangle) \textit{ iff} \\ \langle \_, \tau_1, f_A\rangle \wedge \\ \big(\big((\neg\exists\tau_2 \in \mathrm{T} : \langle \_, \tau_2, f_D\rangle \wedge T2 \geq T1) \wedge \\ (\exists\tau_3 \in \mathrm{T} : \langle w, \tau_3, \varphi\rangle \wedge T3 \geq T1)\big)\end{gathered}$$

□

Inspired by [JS96], we understand institutional power in two forms:

1. Normative power: The power to alter the normative context. Adding norms to it or removing norms, effectively expanding and contracting the normative context. As the framework presented in this section does not account for norm change, normative power do not apply. Therefore, we will define our formal notion of normative power later, in §3.4.
2. Constitutive power: The power to create certain states of affairs by performing specific acts. The typical example is a priest enacting a marriage by the utterance of a particular set of words following a clearly prescribed protocol or ritual.

As we will see in §3.6 the first form of power can be achieved by implementing a filter in the interface that allows agents to modify the normative context. The filter will have a list of agents allowed to alter the normative context. If a request to alter the normative context is performed by an agent on the list, it will be fulfilled, effectively expanding or contracting it. Otherwise the request will be ignored and will have no affect. The list of agents with power to alter the normative state can be represented in the Ontology. In this section we will focus on constitutive power.

Constitutive power creates high level states of affairs from certain low level acts. As an example, a particular actor following a prescribed ritual will count-as a marriage. However, in order for the ritual to be effective it must be performed by an actor with constitutive power, that is, a priest. If the ritual is performed by some other actor without the constitutive power (*e.g.*, during a theatre play or a joke to annoy some work mates) the ritual will have no institutional effect. Our proposal is to support constitutive power via constitutive norms.

Intuitively constitutive power is closely related to constitutive norms, as defined in *Definition* 4 constitutive norms are defined as tuples $c = \langle \gamma_1, \gamma_2, s \rangle$ with two states of the world and a context. In order to effectively apply constitutive power we need to analyse the state of the world $\gamma_1$. When the state of the world is associated to an actor (please, remember our definition of event includes the actor performing it) we have to check if that state is restricted by normative powers. If it is the case, the actor must have constitutive power to bring that state of the world. We can formalize the extension of constitutive norms for supporting constitutive power. For doing it, we will assume there is a list mapping restricted actions to actors. Formally:

**Definition 15** ***(Constitutive Power)***

*Let $c = \langle \gamma_1, \gamma_2, s \rangle$ be a constitutive norm. Where $\gamma_1 \wedge \gamma_2 \in wff(LO)$ that is, they are conjunctions of states of the world.*

*Let $cpow = \langle a, \varrho \rangle$ be an entry in the list of constitutive power permissions, where:*

- *$a \in wff(LO)$ is a state of the world that must be performed by a particular actor or set of actors. Function $a(cpow)$ returns the component a of an entry in the list.*
- *$\varrho$ is the set of actors with constitutive power to perform $a$. Function $\varrho(cpow)$ returns the component $\varrho$ of an entry in the list.*

*Let $\mathcal{P}(r, p)$ be a function that returns true if actor r has constitutive powers on state of the world p. Formally:*

$$\mathcal{P}(r, p) = \exists w \in cpow : a(w) = p \wedge r \in \varrho(w)$$

*Let $H$ be the set of all registered events.*

*We can extend the definition of constitutive norm. Given a constitutive norm $c = \langle \gamma_1, \gamma_2, s \rangle$, $\gamma_1$ entails $\gamma_2$ iff $\gamma_1$ contains a state of the world such that we have registered an event with this state*

| | *Ex Tunc* | *Ex Nunc* |
|---|---|---|
| *Context expansion* | Retroactive Promulgation | Prospective Promulgation |
| *Context contraction* | Annulment | Abrogation |

Figure 3.6: Possible operations on norms

*of the world. In the event, the state of the world is associated to an actor. The actor has constitutive power associated to the state of the world. Formally:*

$$\gamma_1 \textit{ entails } \gamma_2, \textit{formally } \gamma_1 \vDash \gamma_2 \textit{ iff}$$
$$\mathcal{P}(r, \gamma_1) \wedge$$
$$\exists \langle \alpha, t, p \rangle \in \mathcal{E} : \gamma_1 = p \wedge r = \alpha$$

□

*Definition* 15 introduces into our framework the concept of constitutive power. Please notice that we assume function $\mathcal{P}(r, p)$ contains entries for all the possible actions defined in the ontology. A simple enhancement is registering only actions restricted by constitutive power and making the function return $True$ in case the action queried has no entry (*i.e.* it is not restricted by constitutive powers).

Once we have introduced the basic conceptual framework we can extend it to support norm change.

## 3.4 A formal framework for norm change

This section defines the operations required for supporting expansion and contraction of normative contexts. It also depicts the norm life-cycle extension required for supporting these operations.

According to legal literature, one can specify two types of change operations on a normative context: context expansion (adding norms) and context contraction (removing norms; norm updates can be seen as norm removal followed by a norm addition). Each of these operations comes in two forms: *Ex Tunc*(*i.e.*, from the outset) and *Ex Nunc* (*i.e.*, from now on). Both terms are Latin legal terms which are common in law literature. An *Ex Tunc* norm is a norm that retroactively changes the normative consequences (or status) of actions committed prior to the existence of the norm, whereas an *Ex Nunc* norm affects only actions committed after the existence of the norm. To summarize, one can apply four distinct operations to normative contexts, as depicted on *Figure* 3.6.

This section provides a formal definition of the four operations when applied to regulative norms, constitutive norms and institutional powers (including both constitutive powers and normative powers).

### 3.4.1 Regulative norm dynamics

This subsection provides a formal definition of the four operations for norm change when they are applied to regulative norms. A formal definition of the operations is as follows:

***Prospective promulgation*** $+_P$: Introduces a new regulative norm on the normative context. Events happening on the context can instantiate the norm as soon as it has been promulgated. The norm will not check for violations caused by past events, but it can be activated by past events. This is because if a given fact has been made true in the past we assume it to be true until we have prove of the contrary. For instance, if we received the event *bornAt(Spain, Manolete)* in the past we can assume the fact *Manolete* is born in *Spain* still holds in the present. Thus, if a norm applies to (*i.e.*, is activated by) individuals born in Spain, it should apply to the individual *Manolete* even if he was born before the norm promulgation.

Intuitively prospective promulgation of a regulative norm will incur in norm violation only if the event triggering the violation occurs after the operation. Formally:

**Definition 16** (Prospective promulgation of a regulative norm)
*Given the following elements:*

$$\begin{gathered}
\textit{Given the set of times as } [\tau_1, \ldots \tau_k] \in \mathrm{T} \\
\textit{Given the free variable _ which matches any value.} \\
\textit{Given a norm } (n = \langle f_A, f_M, f_D, f_w, w \rangle)
\end{gathered}$$

*We can formalize prospective expansion also known as prospective promulgation:*

$$\begin{gathered}
+_P(n = \langle f_A, f_M, f_D, f_w, w\rangle), \tau_+ \stackrel{\mathsf{def}}{=} \\
Violated(n = \langle f_A, f_M, f_D, f_w, w\rangle) \textit{ iff} \\
\langle _, \tau_1, f_A\rangle \wedge \\
\Big(\big((\neg\exists\tau_2 \in \mathrm{T} : \langle _, \tau_2, f_D\rangle \wedge \tau_2 \geq \tau_1)\wedge \\
(\exists\tau_3 \in \mathrm{T} : \langle w, \tau_3, \neg f_M\rangle \wedge \tau_3 \geq \tau_1 \wedge \tau_3 \geq \tau_+)\big) \\
\vee \\
\big((\exists\tau_2 \in \mathrm{T} : \langle _, \tau_2, f_D\rangle \wedge \tau_2 \geq \tau_1) \wedge \tau_2 \geq \tau_+)\wedge \\
(\neg\exists\tau_3 \in \mathrm{T} : \langle w, \tau_3, f_w\rangle \wedge \tau_3 \geq \tau_1)\big)\Big)
\end{gathered}$$

□

***Retroactive promulgation*** $+_R$: Introduces a new regulative norm on the normative context. Events happening on the context can instantiate the norm as soon as it is promulgated. The norm will check for norm instance activations or violations caused by past events. *Retroactive promulgation* can lead to a massive amount of norm instances being violated (especially if the number of past events is high). Few scenarios should require this operation for normative context modification. In fact, most real-world normative contexts forbid this operation (*e.g.*, most countries forbid retrospective law promulgation on their constitutions and bills of rights).

Intuitively retroactive promulgation of a regulative norm will incur in norm violation even if the event triggering the violation occurs before the operation. Formally:

**Definition 17** (Retroactive promulgation of a regulative norm)
*Given the following elements:*

*Given the set of times as* $[\tau_1, \ldots \tau_k] \in \mathrm{T}$
*Given the free variable _ which matches any value.*
*Given a norm* $(n = \langle f_A, f_M, f_D, f_w, w\rangle)$

*We can formalize retroactive expansion also known as retroactive promulgation:*

$$
\begin{gathered}
+_R(n = \langle f_A, f_M, f_D, f_w, w\rangle), \tau_+ \stackrel{\mathsf{def}}{=} \\
Violated(n = \langle f_A, f_M, f_D, f_w, w\rangle) \textit{ iff} \\
\langle \_, \tau_1, f_A\rangle \wedge \\
\Big(\big((\neg\exists\tau_2 \in \mathrm{T} : \langle \_, \tau_2, f_D\rangle \wedge \tau_2 \geq \tau_1) \wedge \\
(\exists\tau_3 \in \mathrm{T} : \langle w, \tau_3, \neg f_M\rangle \wedge \tau_3 \geq \tau_1)\big) \\
\vee \\
\big((\exists\tau_2 \in \mathrm{T} : \langle \_, \tau_2, f_D\rangle \wedge \tau_2 \geq \tau_1)) \wedge \\
(\neg\exists\tau_3 \in \mathrm{T} : \langle w, \tau_3, f_w\rangle \wedge \tau_3 \geq \tau_1)\big)\Big)
\end{gathered}
$$

□

***Annulment*** $-_R$: Removes a regulative norm from the normative context. All the instances of the norm are removed as well, including violated ones. This implies removing sanctions and repair actions that are yet to be enacted. Repair actions already enacted must be de-enacted, so the agents responsible of enacting repair actions (*i.e.*, manager agent) must be aware of the annulment.

Intuitively annulment of a regulative norm will prevent norm violation even if the event triggering the violation occurs before contraction. Formally:

**Definition 18** (Annulment of a regulative norm)
*Given the following elements:*

*Given the set of times as* $[\tau_1, \ldots \tau_k] \in \mathrm{T}$
*Given the free variable _ which matches any value.*
*Given a norm* $(n = \langle f_A, f_M, f_D, f_w, w\rangle)$

*We can formalize retroactive contraction also known as annulment:*

$$
\begin{gathered}
-_R(n = \langle f_A, f_M, f_D, f_w, w\rangle), \tau_- \stackrel{\mathsf{def}}{=} \\
Violated(n = \langle f_A, f_M, f_D, f_w, w\rangle) \textit{ iff } false \wedge \\
compensated(n = \langle f_A, f_M, f_D, f_w, w\rangle) \textit{ iff } repaired(n = \langle f_A, f_M, f_D, f_w, w\rangle) \wedge \\
repaired(n = \langle f_A, f_M, f_D, f_w, w\rangle) \textit{ iff } false \wedge \\
fulfilled(n = \langle f_A, f_M, f_D, f_w, w\rangle) \textit{ iff } false
\end{gathered}
$$

□

Please notice that our norm life supports both violated and repaired norm instances, and they receive a different treatment. Specifically, repaired norm instances are compensated when norm is abrogated. It is worth to remark that we apply norm compensations

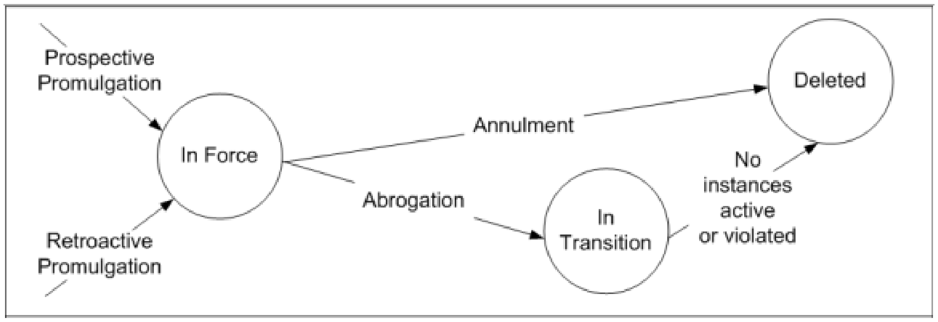



Figure 3.7: Extending the life cycle of regulative norms

before removing norm reparations. We will introduce a formal definition of compensated norm instances once we finish extending our norm life-cycle.

***Abrogation*** $-_P$: Tags a norm from the normative context as being *In transition*. The norm can not be instantiated anymore. Instances of the norm remain in the normative context as long as they are not in a terminal state (that is, either fulfilled or repaired). Once all norm's instances have reached a terminal state, the norm is removed from the system along with its instances.

Intuitively abrogation of a norm will prevent norm violation only if the event triggering the violation occurs after the operation. Please notice that our norm life cycle also supports violated and repaired norm instances. That is why we keep norm instances which have been violated before context contraction and are not repaired yet. Formally:

**Definition 19** (Abrogation of a regulative norm)
*Given the following elements:*

*Given the set of times as* $[\tau_1, \ldots \tau_k] \in \mathrm{T}$
*Given the free variable _ which matches any value.*
*Given a norm* $(n = \langle f_A, f_M, f_D, f_w, w\rangle)$

*We can formalize prospective contraction also known as abrogation:*

$$\begin{gathered}
-_P(n = \langle f_A, f_M, f_D, f_w, w\rangle), \tau_- \stackrel{\text{def}}{=} \\
Violated(n = \langle f_A, f_M, f_D, f_w, w\rangle) \text{ iff} \\
\langle _, \tau_1, f_A\rangle \wedge \\
\Big(((\neg\exists\tau_2 \in \mathrm{T} : \langle _, \tau_2, f_D\rangle \wedge \tau_2 \geq \tau_1) \wedge \\
(\exists\tau_3 \in \mathrm{T} : \langle w, \tau_3, \neg f_M\rangle \wedge \tau_3 \geq \tau_1 \wedge \tau_3 \leqslant \tau_-)) \\
\vee \\
((\exists\tau_2 \in \mathrm{T} : \langle _, \tau_2, f_D\rangle \wedge \tau_2 \geq \tau_1) \wedge \tau_2 \leqslant \tau_-) \wedge \\
(\neg\exists\tau_3 \in \mathrm{T} : \langle w, \tau_3, f_w\rangle \wedge \tau_3 \geq \tau_1))\Big)
\end{gathered}$$

□

In order to effectively support these operations we have to extend our basic regulative norm life-cycle introduced in *Definition* 7. *Figure* 3.7 depicts the new norm lifeycle adding the following states:

- In force: Once a regulative norm has been promulgated (either in a prospective or retroactive form) it achieves an *In force* state. From this point, the regulative norm can be effectively activated if its activation condition is met. In some scenarios, regulative norms introduced in the system off-line (*i.e.*, they are already there when the system starts its execution) are by default in this state. In other scenarios they can lack this state (*i.e.*, be in *Deleted* stated instead) and are moved to it by institutional agents in case they consider they are beneficial for the overall goals of the system. This will effectively provide agents with a pool of norms that can be promulgated (either in a prospective or retroactive form) if required. A mixture of both approaches is also possible, where system designers put some important norms *In force* since the beginning of system's execution, leaving a second set of norms in the pool of norms that can be put into force by institutional agents.
- In transition: Abrogated regulative norms go into this state. It means the regulative norm can not be instantiated anymore. However, instances of the norm that have been already instantiated (*i.e.*, they are in *active* or *violated* state) remain in the system. Once these instances change to *Fulfilled* state the norm *in transition* can effectively move on to *Deleted* state.
- Deleted: As stated before, abrogated regulative norms with no active instances are moved to this state. Annulled regulative norms are also moved to this state, no matter if they contain active instances or not. Therefore, active instances of annulled norms are removed from the system. Mechanisms based on the already available violation handling can be defined to compensate for violated instances of annulled norms that have been repaired (*e.g.*, if an agent pays a fine for violating a norm, and then the norm is annulled, return the amount paid to the agent).

In *Definition* 6 we introduced the set of states on our regulative norm life-cycle. That is, the set of states where norm instances transition. In order to support the new norm life-cycle we have to extend these states. Informally, we have to add compensated set of norm instances for the compensations generated by regulative norm annulment. Formally:

**Definition 20** (Dynamic Normative States)

$$IS \textit{ is an instantiation set}$$

$$FS \textit{ is a fulfilment set}$$

$$VS \textit{ is a violation set}$$

$$RS \textit{ is a repairment set}$$

$$CS \textit{ is a compensated set}$$

$$\textit{Each of these sets contains norm instances } \{\langle n_i, \Theta_j \rangle, ..., \langle n_{i'}, \Theta_{j'} \rangle\}$$

$$\textit{The normative state is a tuple formed by these sets } NS = \{\langle IS, FS, VS, RS, CS \rangle\}$$

□

In *Definition* 7 we introduced the operational definition of the regulative norm instance life cycle. By adding a new state (*i.e.* compensated) with the corresponding set of norm instances (*i.e.* $CS$) we have to extend the operational definition as well. Formally:

**Definition 21** (Dynamic Regulative Norm Lifecycle) *Let $ni = \langle n, \Theta \rangle$ be a regulative norm instance, such that $n = \langle f_A, f_M, f_D, w \rangle$, and $s$ be a state of the world with an expansion $F(s)$. Then we extend the lifecycle for a norm instance $ni$ by the following normative state predicates:*

$activated(ni) \Leftrightarrow \exists f \in F(s), \Theta(f_A) \equiv f$
$maintained(ni) \Leftrightarrow \exists \Theta', \exists f \in F(s), \Theta'(f_M) \equiv f \wedge \Theta' \subseteq \Theta$
$deactivated(ni) \Leftrightarrow \exists \Theta', \exists f \in F(s), \Theta'(f_D) \equiv f \wedge \Theta' \subseteq \Theta$
$instantiated(ni) \Leftrightarrow ni \in IS$
$violated(ni) \Leftrightarrow ni \in VS$
$fulfilled(ni) \Leftrightarrow ni \in FS$
$repaired(ni, ni') \Leftrightarrow \langle ni, ni' \rangle \in RS$
$ni' \in CS \Leftrightarrow \exists ni : \langle ni, ni' \rangle \in RS \wedge \exists \Theta : ni = \langle n, \theta \rangle \wedge -_R(n)$
$compensated(ni') \Leftrightarrow ni' \in CS$

*Where $IS$ is the instantiation set, $FS$ is the fulfillment set, $VS$ is the violation set, and $RS$ is the set of those norm instances $ni'$ that have repaired a norm instance $ni$. Finally CS is the set of norm instances $ni'$ that have repaired a norm instance $ni$ where $ni$ is an instantiation of a norm $n$ that has been annulled.*

□

In the base framework introduced in §3.3 norm instances have states, because they transition from one to another as events are processed. However, regulative norms lack the definition of state, because they have only one possible state (the equivalent to our *In force* state). We can extend the base framework with the notion of regulative norm state. Informally we have to account for the following states:

- In force ($N_F$): The set of regulative norms in force. An initial set of regulative norms and the promulgated ones.
- In transition ($N_T$): Set of abrogated regulative norms with violated or active norm instances.
- Deleted ($N_D$): Set of abrogated regulative norms without violated or active norm instances. Also includes the set of annulled norms.

For doing it we extend the base definition of normative monitor in *Definition* 10, formally:

**Definition 22** (Dynamic Normative Monitor) *A Dynamic Normative Monitor $M_N$ for a set of initial regulative norms $N$ is a tuple $M_N = \langle \langle N_F, N_T, N_D \rangle, S, IS, VS, FS, RS, CS, H \rangle$. Where:*

- $S = \mathcal{P}(O)$.
- $IS = \mathcal{P}(N \times S \times Dom(S))$.
- $VS = \mathcal{P}(N \times S \times Dom(S))$.
- $FS = \mathcal{P}(N \times S \times Dom(S))$.
- $RS = \mathcal{P}(N \times S \times Dom(S))$.
- $CS = \mathcal{P}(N \times S \times Dom(S))$.
- *$H \in E$ is the history of events that has been observed by the monitor.*
- $N_F = N \cup (n : +_P(n) \vee +_R(n))$.
- $N_T = n \in N_F : -_P(n) \wedge \exists n_i, \Theta : ni = \langle n, \theta \rangle \wedge (ni \in IS \vee ni \in VS)$.

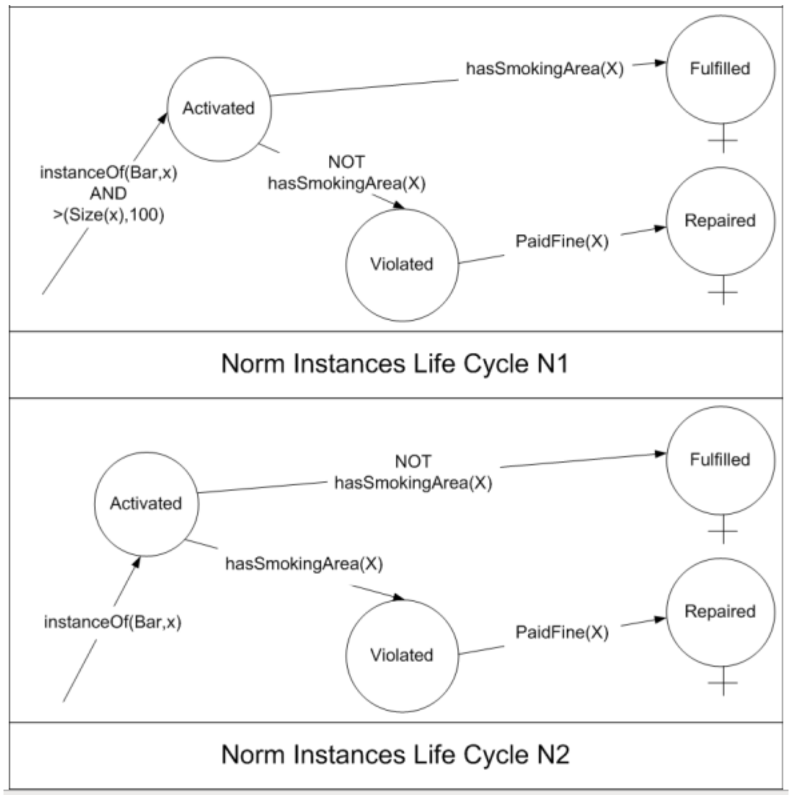



Figure 3.8: Example of the execution of regulative norms N1 and N2

- $N_D = n \in N_F : -_R(n) \cup n \in N_T : \neg\exists n_i, \Theta : ni = \langle n, \theta \rangle \wedge (ni \in IS \vee ni \in VS)$.

□

Figure 3.8 shows examples of some operations on a normative context. Specifically it shows how the regulative norm (*N1*) and the amendment (*N2*) are activated, fulfilled, violated or repaired depending on the events happening on the environment.

Basically *N1* is activated as soon as bar instances with more than $100m^2$ are detected on the environment. Please notice how both prospective and retroactive norm promulgation check for past events. In this case, it is because if in the past a bar instance had more than $100m^2$ we assume this fact to be still true if we have not prove of the contrary. Once the regulative norm is promulgated it instantiates. The instance goes to fulfilled state if the bar has an isolated area for smoking customers, and to violated state if it does not. If the regulative norm is violated, violation can be repaired by creating an isolated area for smokers. This action will take the norm to repaired state.

Then, regulative norm *N1* is abrogated and *N2* promulgated. Once promulgated *N2* instantiates as soon as bar instances are detected, no size constraints are to be met. The norm instance is fulfilled if the bar has no smoking area, and violated if it does. If the norm is violated, violation can be repaired by removing the isolated smoking area from the bar. As norm *N1* has been abrogated, Bars that are on violated state on instances of *N1* have to perform their repair actions in order to move to repaired state even if *N1* has been already abrogated and *N2* promulgated. It would be the case if norm *N1* was annulled instead of abrogated as annulment would force norm violations to be removed from the system.

This simple scenario allows us to reinforce the idea system designers need to be very careful when using regulative norm abrogation. Bars that have spent money in creating areas for smokers due to the promulgation of *N1*, have to spend money again for removing these areas once norm *N2* is promulgated. Otherwise they would be violating *N2*. Therefore, in order to achieve a fair regulative norm promulgation, system designers should include a reward for bars that fulfilled norm *N1* by creating areas for smokers when promulgating norm *N2*.

### 3.4.2 Constitutive norm, constitutive power and normative power dynamics

We can apply the same operations to constitutive norms as introduced in *Definition* 4. Intuitively:

- Prospective expansion of a constitutive norm implies the brute facts will count as institutional facts as long as they occur after expansion.
- Retroactive expansion of a constitutive norm implies the brute facts will count as institutional facts even if they occur before expansion.
- Prospective contraction of a constitutive norm implies the brute facts will not count as institutional facts as long as they occur before contraction.
- Prospective contraction of a constitutive norm implies the brute facts will not count as institutional even if they occur before contraction.

Formally:

**Definition 23** (Operations on constitutive norms)

*In our proposal we observe brute facts via events. The formal concept of event $\langle \alpha, t, p \rangle$ is tightly coupled with the concept of time 't'. We can formalize constitutive norm expansion and contraction operations from this set of elements:*

*Given the set of times as* $[\tau_1, \ldots \tau_k] \in \mathrm{T}$
*Given the free variable _ which matches any value.*
*Given a constitutive norm* $c = \langle \gamma_1, \gamma_2, s \rangle$
*And the entailment definition associated* $\gamma_1$ *entails* $\gamma_2$*, formally* $\gamma_1 \vDash \gamma_2$
*Given an event* $\langle \alpha, t, p \rangle$ *and the set of all events* $\mathcal{E}$

*We can formalize prospective expansion also known as prospective promulgation of a constitutive norm:*

$$+_P(c = \langle \gamma_1, \gamma_2, s \rangle), \tau_+ \stackrel{\text{def}}{=}$$
$\gamma_1$ *entails* $\gamma_2$*, formally* $\gamma_1 \vDash \gamma_2$ *iff*
$$\exists \langle \alpha, t, p \rangle \in \mathcal{E} : \gamma_1 = \alpha \wedge t \geqslant \tau_+$$

*We can formalize retroactive expansion also known as retroactive promulgation of a constitutive norm:*

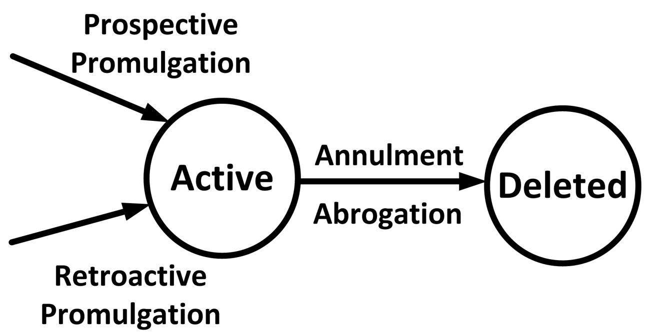



Figure 3.9: Extending the life cycle of constitutive norms and constitutive powers

$$+_R(c = \langle \gamma_1, \gamma_2, s\rangle), \tau_+ \stackrel{\text{def}}{=}$$
$$\gamma_1 \textit{ entails } \gamma_2, \textit{formally } \gamma_1 \vDash \gamma_2 \textit{ iff}$$
$$\exists \langle \alpha, t, p\rangle \in \mathcal{E} : \gamma_1 = \alpha$$

*We can formalize prospective contraction also known as abrogation of a constitutive norm:*

$$-_P(c = \langle \gamma_1, \gamma_2, s\rangle), \tau_- \stackrel{\text{def}}{=}$$
$$\gamma_1 \textit{ entails } \gamma_2, \textit{formally } \gamma_1 \vDash \gamma_2 \textit{ iff}$$
$$\exists \langle \alpha, t, p\rangle \in \mathcal{E} : \gamma_1 = \alpha \wedge t \leqslant \tau_-$$

*We can formalize retroactive contraction also known as annulment of a constitutive norm:*

$$-_R(c = \langle \gamma_1, \gamma_2, s\rangle), \tau_- \stackrel{\text{def}}{=}$$
$$\gamma_1 \textit{ entails } \gamma_2, \textit{formally } \gamma_1 \vDash \gamma_2 \textit{ iff } false$$

□

When defining dynamics for constitutive powers, the idea is exactly the same as in constitutive norms. All we have to do is match the event with the brute fact with the actor uttering it and the time when the power was added or removed from the normative context. For simplicity, we will skip the formal definition, as the analogous definition of the operations for constitutive norms has been already provided.

In the base framework presented in §3.3 constitutive norms have no life-cycle because there is only one possible state. The same stands for constitutive powers. In order to effectively support these operations we have to extend our constitutive norm and normative power life-cycle, as depicted in *Figure* 3.7, adding the following states:

- Active ($C_A$): Once a constitutive norm has been promulgated (either in a prospective or retroactive form) it achieves an *Active* state. From this point, the constitutive norm starts interpreting brute facts. The state contains both the constitutive norm and a time stamp. The same holds for constitutive powers with the set $P_A$.
- Deleted ($C_D$): Once a constitutive norm has been annulled or abrogated it achieves a *Deleted* state. From this point, the constitutive norm will not interpret brute facts

anymore. The state contains both the constitutive norm and a time stamp. The same holds for constitutive powers with the set $P_D$.

To add the states we extend the Dynamic Normative Monitor in *Definition* 22, formally:

**Definition 24** (Extended Dynamic Normative Monitor) *A Dynamic Normative Monitor $M_N$ for a set of initial regulative norms $N$, a set of initial constitutive norms $C$, and a set of initial constitutive powers $P_C$ is a tuple $M_N = \langle\langle N_F, N_T, N_D, C_A, C_D\rangle, S, IS, VS, FS, RS, CS, H, \langle P_A, P_D\rangle\rangle$. Where:*

- $S = \mathcal{P}(O)$.
- $IS = \mathcal{P}(N \times S \times Dom(S))$.
- $VS = \mathcal{P}(N \times S \times Dom(S))$.
- $FS = \mathcal{P}(N \times S \times Dom(S))$.
- $RS = \mathcal{P}(N \times S \times Dom(S))$.
- $CS = \mathcal{P}(N \times S \times Dom(S))$.
- *$H \in E$ is the history of events that has been observed by the monitor.*
- $N_F = N \cup (n : +_P(n) \vee +_R(n))$.
- $N_T = n \in N_F : -_P(n) \wedge \exists n_i, \Theta : ni = \langle n, \theta\rangle \wedge (ni \in IS \vee ni \in VS)$.
- $N_D = n \in N_F : -_R(n) \cup n \in N_T : \neg\exists n_i, \Theta : ni = \langle n, \theta\rangle \wedge (ni \in IS \vee ni \in VS)$.
- *Let the actual time be $t$.*
- $C_A = (\langle c, 0\rangle : c \in C) \cup (\langle c, 0\rangle : +_R(c)) \cup (\langle c, t\rangle : +_P(c))$
- $C_D = (\langle c, 0\rangle : -_R(c)) \cup (\langle c, t\rangle : -_P(c))$
- $P_A = (\langle p, 0\rangle : p \in P_C) \cup (\langle p, 0\rangle : +_R(p)) \cup (\langle p, t\rangle : +_P(p))$
- $P_D = (\langle p, 0\rangle : -_R(p)) \cup (\langle c, t\rangle : -_P(p))$

□

Finally, we can extend the *Labelled Transition System* for a Normative Monitor definition. Intuitively a constitutive norm will activate only under the following conditions:

- The brute fact $\gamma_1$ holds true.
- $\gamma_1$ has happened on a time when the constitutive norm is active, according to the set $C_A$.
- $\gamma_1$ has been uttered by actor with constitutive power.
- The constitutive power is active on the time when $\gamma_1$ holds true, according to $P_A$.

Formally:

**Definition 25** (Constitutive norm activation)

*Let $utters(\gamma_1, \alpha)$ be a function that returns true if the state of the world $\gamma_1$ is uttered by actor $\alpha$, and false otherwise.*

$$\frac{\Theta(\gamma_1) \equiv \langle \alpha, t, p\rangle \qquad \Theta(\gamma_2) \notin s\langle\gamma_1, \gamma_2, s_i\rangle \in C \qquad f \in F(s) \qquad s_i \subseteq s \qquad c = \langle\gamma_1, \gamma_2, s_i\rangle \wedge \exists < c, t' > \in C_A : t' \geq t \wedge utters(\gamma_1, \alpha) \wedge \exists < p, t'' > \in P_A : t'' \geq t \wedge p = \langle\gamma_1, \alpha\rangle}{\langle s\rangle \rhd \langle s \cup \{\Theta(\gamma_2)\}\rangle}$$

□

Once we have introduced the operations to update the normative context we can provide a definition of normative power. Normative power is the power to alter the normative context, that is, power to perform the operations we have introduced on this section. The operations are requested by the actors in the institution and affect the norms in the normative context. Therefore, normative power is a list mapping actors to norms, $< a, n >$ meaning actor $a$ has normative power to modify norm $n$. Norm stands for regulative norm, constitutive norm, constitutive power and even normative power itself. Therefore, we support an actor granting normative power to another actor on a particular set of norms. Modify stands for applying any operation defined in this section to the norm. We could easily extend the concept of normative power to differentiate between the operations (*i.e.* an actor can abrogate a particular norm but not annul it) but we consider such fine-grained permissions are not required in the scope of our framework. Formally, we define normative power as:

**Definition 26** (Normative power)
*Let:*

$a$ *be an actor defined in the ontology:* $a \in O$
$P_N$ *denotes the set of all normative powers.*
*n be a norm:* $n \in N \wedge N = N_F \cup N_T \cup N_D \cup C_A \cup C_D \cup P_A \cup P_D \cup P_N$

$a$ *has normative power to modify norm* $n$ *is formally denoted with the tuple:*

$$< a, n >$$

*Where:*

$P_N$ *denotes the set of all normative powers:* $P_N = \langle < a_1, n_1 >, < a_1, n_2 > \cdots < a_n, n_n > \rangle$
*We can use the free variable _ to denote any actor or any norm.*
$< a, \_ >$ *means actor* $a$ *has normative power on any norm.*
$< \_, n >$ *means any actor has normative power on norm* $n$.

□

Intuitively, normative power dynamics account for adding normative powers when expansion operations are performed on the set $P_N$ (either prospectively or retroactively) and removing normative powers when contraction operations are performed on $P_N$ (either prospectively or retroactively). In the scope of our framework we will not distinguish between prospective and retroactive operations *w.r.t.* normative powers. Therefore, prospective promulgation of a normative power $\phi$ (denoted as $+_P(\phi)$) is equivalent to the retroactive promulgation of a normative power $\phi$ (denoted as $+_R(\phi)$). The same stands for abrogation of a normative power $\phi$ (denoted as $-_P(\phi)$) and annulment of a normative power $\phi$ (denoted as $-_R(\phi)$). In §3.6 we will see our framework can be easily extended for accounting for prospective and retroactive operations on normative powers. Formally:

**Definition 27** (Normative power dynamics) *Let:*

$+_P(\langle a,n\rangle)$ *denote prospective promulgation of normative power to actor $a$ on norm $n$*
$+_R(\langle a,n\rangle)$ *denote retroactive promulgation of normative power to actor $a$ on norm $n$*
$-_P(\langle a,n\rangle)$ *denote abrogation of normative power to actor $a$ on norm $n$*
$-_R(\langle a,n\rangle)$ *denote annulment of normative power to actor $a$ on norm $n$*

*We support only prospective operations on normative powers, therefore:*

$$+_P(\langle a,n\rangle) \stackrel{\text{def}}{=} +_R(\langle a,n\rangle)$$
$$-_P(\langle a,n\rangle) \stackrel{\text{def}}{=} -_R(\langle a,n\rangle)$$

*To provide a clearer definition closer to natural language we denote:*

$+Pow(a,n)$ *as the prospective promulgation of normative power to actor $a$ on norm $n$*
$-Pow(a,n)$ *as the abrogation of normative power to actor $a$ on norm $n$*

□

We define normative power dynamics as:

**Definition 28** (Extended Dynamic Normative Monitor with normative power) *A Dynamic Normative Monitor $M_N$ for a set of initial regulative norms $N$, a set of initial constitutive norms $C$, a set of initial constitutive powers $P_C$ and a set of initial normative powers $P_P$ is a tuple $M_N = \langle\langle N_F, N_T, N_D, C_A, C_D\rangle, S, IS, VS, FS, RS, CS, H, \langle P_A, P_D\rangle, P_N\rangle$. Where:*

- $S = \mathcal{P}(O)$.
- $IS = \mathcal{P}(N \times S \times Dom(S))$.
- $VS = \mathcal{P}(N \times S \times Dom(S))$.
- $FS = \mathcal{P}(N \times S \times Dom(S))$.
- $RS = \mathcal{P}(N \times S \times Dom(S))$.
- $CS = \mathcal{P}(N \times S \times Dom(S))$.
- $H \in E$ *is the history of events that has been observed by the monitor.*
- $N_F = N \cup (n : +_P(n) \vee +_R(n))$.
- $N_T = n \in N_F : -_P(n) \wedge \exists n_i, \Theta : ni = \langle n, \theta\rangle \wedge (ni \in IS \vee ni \in VS)$.
- $N_D = n \in N_F : -_R(n) \cup n \in N_T : \neg\exists n_i, \Theta : ni = \langle n, \theta\rangle \wedge (ni \in IS \vee ni \in VS)$.
- *Let the actual time be $t$.*
- $C_A = (\langle c, 0\rangle : c \in C) \cup (\langle c, 0\rangle : +_R(c)) \cup (\langle c, t\rangle : +_P(c))$
- $C_D = (\langle c, 0\rangle : -_R(c)) \cup (\langle c, t\rangle : -_P(c))$
- $P_A = (\langle p, 0\rangle : p \in P_C) \cup (\langle p, 0\rangle : +_R(p)) \cup (\langle p, t\rangle : +_P(p))$
- $P_D = (\langle p, 0\rangle : -_R(p)) \cup (\langle c, t\rangle : -_P(p))$
- $P_N = (\phi : \phi \in P_P \cup \phi : +_P(\phi) \cup \phi : +_R(\phi)) - (\phi : -_P(\phi) \cup \phi : -_R(\phi))$

□

Once we have completed the formal definition of our framework and the operations to update the normative context we will proceed by describing the operational semantics of our framework. We do so by introducing the pseudocode of the algorithms required for fulfilling each normative context operation from a monitoring perspective.

## 3.5 Implementing Norm change support

In this section we present the operational semantics of our framework. We do so by introducing the pseudocode of the algorithms required for fulfilling each normative context operation form a monitoring perspective. The operational semantics will take care of some aspects we have not covered in the formalisation provided in §3.4 such as the operational description of how to maintain each set of norm instances.

If norms are represented as rules, then rule change can be represented as non-monotonic inference. According to [GR08a], changing a Normative System would amount to adding new rules or removing the existing ones. The formalism presented in Section 3.3, based on rule creation from normative specifications, allows us to define the operationalisation of norm change, in its four forms, as extensions of the main monitoring process. By using all or some of the labels described in *Definition* 11, we can constrain the exact normative-related actions that the monitor will be able to carry out, and thus we can define and create, at runtime, diverse monitoring *contexts*.

The ability to create a different monitoring context is due to the fact that some norm changes can be retroactive. Thus the only way to generate a normative state compliant with the one we had before the norm change is to analyse the full stream of events generated since the beginning of the monitoring process. Such a monitoring context can be created at runtime by using two constructs from the monitoring formalism: the *normative monitor* and the labels of its *labelled transition system*. With the first one we can create a new monitor, specific to a set of norms (the ones to be added), and by constraining the labels we can control the level of retroactivity of the norm change type.

How these constructs have to be used depends on each type of norm change, as defined in §3.4. As seen on *Figure* 3.6 there are a total of four possible operations on a normative context, formalised by the algorithms presented in this section. First we will introduce the application of the operations to regulative norms. Then, to constitutive norms and normative powers.

### 3.5.1 Implementing Norm change support for regulative norms

**Algorithm 1** Prospective Promulgation of *PPNorm*

**Require:** $PPNorm = \langle f_A, f_M, f_D, f_w, w \rangle$
**Require:** $M_N = \langle\langle N_F, N_T, N_D, C_A, C_D\rangle, S, IS, VS, FS, RS, CS, H, \langle P_A, P_D\rangle, P_N\rangle$
**Require:** $PPNorm \notin N_F$
  $M_{N'} = \langle\langle N_F \cup \{PPNorm\}, N_T, N_D, C_A, C_D\rangle, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, H, \langle P_A, P_D\rangle, P_N\rangle$
  $LTS_{M_{N'}} = \{\Gamma, \{nii\}, \triangleright\}$
  $engine.create(LTS_{M_{n'}})$
  $it = E.iterator$
  **while** $it.hasNext$ **do**
    $engine.insert(it.next)$
    $engine.infer$
  $M_N.IS = M_N.IS \cup M_{N'}.IS$

*Algorithm* 1 models the prospective promulgation of the regulative norm $PPNorm$. The idea is creating an auxiliary monitor $M_{N'}$ with the current set of norms and the promulgated regulative norm. We run the operational rules on the auxiliary monitor, effectively inferring the set of norm instances and their respective states. As the promulgated norm can be instantiated by past events (but not violated,repaired or fulfilled) we merge the auxiliary monitor and the main one by inserting in the main monitor only the norm instances which are in instantiated state (denoted as $M_{N'}.IS$) according to the auxiliary monitor.

**Algorithm 2** Retroactive Promulgation of *RPNorm*

**Require:** $RPNorm = \langle f_A, f_M, f_D, f_w, w \rangle$
**Require:** $M_N = \langle \langle N_F, N_T, N_D, C_A, C_D \rangle, S, IS, VS, FS, RS, CS, H, \langle P_A, P_D \rangle, P_N \rangle$
**Require:** $RPNorm \notin N_F$
  $M_{N'} = \langle \langle N_F \cup \{RPNorm\}, N_T, N_D, C_A, C_D \rangle, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, H, \langle P_A, P_D \rangle, P_N \rangle$
  $LTS_{M_{N'}} = \{\Gamma, \{nii, niv, nif, nir\}, \triangleright\}$
  $engine.create(LTS_{M_{n'}})$
  $it = E.iterator$
  **while** $it.hasNext$ **do**
    $engine.insert(it.next)$
    $engine.infer$
  $M_N.IS = M_N.IS \cup M_{N'}.IS$
  $M_N.VS = M_N.VS \cup M_{N'}.VS$
  $M_N.FS = M_N.FS \cup M_{N'}.FS$
  $M_N.RS = M_N.RS \cup M_{N'}.RS$

*Algorithm* 2 models the retroactive promulgation of the regulative norm $RPNorm$. The idea is creating an auxiliary monitor $M_{N'}$ with the current set of norms and the promulgated regulative norm. We run the operational rules on the auxiliary monitor, effectively inferring the set of norm instances and their respective states. As the promulgated norm can be in any state (*e.g.,* instantiated, violated, fulfilled or repaired) due to past events we merge the auxiliary monitor and the main one by performing the following sequence of operations:

- Inserting in the main monitor the norm instances which are in instantiated state (denoted as $M_{N'}.IS$) according to the auxiliary monitor.
- Inserting in the main monitor the norm instances which are in violated state (denoted as $M_{N'}.VS$) according to the auxiliary monitor.
- Inserting in the main monitor the norm instances which are in fulfilled state (denoted as $M_{N'}.FS$) according to the auxiliary monitor.
- Inserting in the main monitor the norm instances which are in repaired state (denoted as $M_{N'}.RS$) according to the auxiliary monitor.

*Algorithm* 3 models the annulment of the regulative norm $AnNorm$. The idea is removing from the monitor's memory all references to the annulled regulative norm. Norm instances in either instantiated, violated or fulfilled state are directly removed. In the case of repaired instances, apart from performing the removal operation, we also notify a manager (*i.e.* an institutional agent responsible of sanctions) the repaired norm instances affected by the annulment. This way, the manager can compensate the sanctions paid by agents when repairing the annulled regulative norm (*e.g.,* return fines paid or take actors out of jail) effectively undoing the sanctions associated to norm repairing.

*Algorithm* 4 models the abrogation of the regulative norm $AbNorm$. The operation consists in three main steps. First, we remove fulfilled and repaired instances of the regulative norm from the monitor's memory. Then, we mark the norm to be in transition, effectively

**Algorithm 3** Annulment of *AnNorm*

**Require:** $AnNorm = \langle f_A, f_M, f_D, f_w, w\rangle$
**Require:** $M_N = \langle\langle N_F, N_T, N_D, C_A, C_D\rangle, S, IS, VS, FS, RS, CS, H, \langle P_A, P_D\rangle, P_N\rangle$
**Require:** $AnNorm \in N_F$
  **for all** $ni \in IS$ **do**
    **if** $ni.norm == AnNorm$ **then**
      $IS = IS - ni$
  **for all** $ni \in VS$ **do**
    **if** $ni.norm == AnNorm$ **then**
      $VS = VS - ni$
  **for all** $ni \in FS$ **do**
    **if** $ni.norm == AnNorm$ **then**
      $FS = FS - ni$
  **for all** $\langle ni, ni'\rangle \in RS$ **do**
    **if** $ni.norm == AnNorm$ **then**
      $RS = RS - \langle ni, ni'\rangle$
      $CS = CS + ni'$
  $N_F = N_F - AnNorm$
  $N_D = N_D + AnNorm$

blocking its instantiation. It means the monitor will not infer new instances of the regulative norm from this point of time. Therefore, no new norm instances will be in instantiated state. An easy way to accomplish this is to keep a list of regulative norms in transition on the monitor's knowledge base. Instances that are already in instantiated state will transition to fulfilled or violated states as a consequence of agent's actions. Instances that are already in violated state will transition to repaired state as a consequence of agent's actions. Finally, we perform a periodic check on the norm instances. We query monitor's memory for fulfilled and repaired instances of the norm. If we find any of them, we remove the instance form memory. Otherwise, it means the norm can be deleted from the set of norms in the monitor, effectively finishing the abrogation operation.

Once we have introduced the algorithms for regulative norms, we can proceed to constitutive norms and normative powers.

### 3.5.2 Implementing Norm change support for constitutive norms and normative powers

As seen in *Definition* 25, modifying constitutive norms (either prospectively or retroactively) will account for a modification of $C_A$ and $C_D$ sets and running the production rules used in the monitor. Therefore, the operational semantics is contained in the production rules, and no meta-algorithm is required (except for modifying sets $C_A$ and $C_D$, which is trivial). Same stands for constitutive powers with sets $P_A$ and $P_D$. An informal definition of the procedures associate to each operation is as follows:

**Prospective Promulgation of a constitutive norm** $c$: Add $\langle c, t\rangle$ to the set of active constitutive norms $C_A$ where $t$ is the actual time. $C_A = C_A \cup \langle c, t\rangle$. As new events are processed, institutional facts will be entailed by the new norm. If $\langle c, \_\rangle$ is in the set $C_D$ remove it from this set.

**Retroactive Promulgation of a constitutive norm** $c$: Add $\langle c, 0\rangle$ to the set of active constitutive norms $C_A$. $C_A = C_A \cup \langle c, 0\rangle$. Run the rules engine to process old events, entailing

**Algorithm 4** Abrogation of *AbNorm*

**Require:** $AbNorm = \langle f_A, f_M, f_D, f_w, w \rangle$
**Require:** $M_N = \langle \langle N_F, N_T, N_D, C_A, C_D \rangle, S, IS, VS, FS, RS, CS, H, \langle P_A, P_D \rangle, P_N \rangle$
**Require:** $AbNorm \in N_F$
  **for all** $ni \in FS$ **do**
    **if** $ni.norm == AbNorm$ **then**
      $FS = FS - ni$
  **for all** $\langle ni, ni' \rangle \in RS$ **do**
    **if** $ni.norm == AbNorm$ **then**
      $RS = RS - \langle ni, ni' \rangle$
  {At this point, $AbNorm$ becomes $InTransition$}
  $T_S = T_S + AbNorm$
  **while** $AbNorm \in N$ **do**
    $deleteNorm = true$
    **for all** $ni \in IS$ **do**
      **if** $ni.norm == AbNorm$ **then**
        $deleteNorm = false$
    **for all** $ni \in VS$ **do**
      **if** $ni.norm == AbNorm$ **then**
        $deleteNorm = false$
    **if** $deleteNorm$ **then**
      **for all** $ni \in FS$ **do**
        **if** $ni.norm == AbNorm$ **then**
          $FS = FS - ni$
      **for all** $\langle ni, ni' \rangle \in RS$ **do**
        **if** $ni.norm == AbNorm$ **then**
          $RS = RS - \langle ni, ni' \rangle$
      $N_F = N_F - AnNorm$
      $N_D = N_D + AnNorm$
    $sleep(sometime)$

institutional facts via the new norm. If $\langle c, _ \rangle$is in the set $C_D$ remove it from this set.

**Abrogation of a constitutive norm** $c$: Remove $\langle c, \_ \rangle$ from the set of active constitutive norms $C_A$. $C_A = C_A - \langle c, \_ \rangle$. As new events are processed, institutional facts entailed by the old norm are blocked. Add $\langle c, t \rangle$ to the set $C_D$ where $t$ is the actual time.

**Annulment of a constitutive norm** $c$: Remove $\langle c, \_ \rangle$ from the set of active constitutive norms $C_A$. $C_A = C_A - \langle c, \_ \rangle$. Run the rules engine to process old events, blocking institutional facts entailed by the old norm. Add $\langle c, 0 \rangle$ to the set $C_D$.

**Prospective Promulgation of a constitutive power** $p$: Add $\langle p, t \rangle$ to the set of active constitutive powers $P_A$ where $t$ is the actual time. $P_A = P_A \cup \langle p, t \rangle$. Constitutive power applies to new utterances received as events. If $\langle p, \_ \rangle$ is in the set $P_D$ remove it from this set.

**Retroactive Promulgation of a constitutive power** $p$: Add $\langle p, 0 \rangle$ to the set of active constitutive powers $P_A$. $P_A = P_A \cup \langle p, 0 \rangle$. Check history of utterances received as events, constitutive power applies to all of them. If $\langle p, \_ \rangle$ is in the set $P_D$ remove it from this set.

**Abrogation of a constitutive power** $p$: Remove $\langle p, \_ \rangle$ from the set of active constitutive powers $P_A$. $P_A = P_A - \langle p, \_ \rangle$. Constitutive power is blocked for new utterances received as events. Add $\langle p, t \rangle$ to the set $P_D$ where $t$ is the actual time.

**Annulment of a constitutive power** $p$: Remove $\langle p, \_ \rangle$ from the set of active constitutive powers $P_A$. $P_A = P_A - \langle p, \_ \rangle$. Check history of utterances received as events, constitutive power is blocked on all of them. Add $\langle p, 0 \rangle$ to the set $P_D$.

Modifying normative powers (either prospectively or retroactively) will account for modifying the list of normative powers inside the monitor's memory. As this operation is trivial, we will not introduce an algorithm for depicting it. An informal definition of the procedures associate to each operation is as follows:

**Prospective Promulgation of a normative power** $\phi = \langle a, n \rangle$: Add $\phi$ to the set of normative powers $P_N$. $P_N = P_N \cup \phi$. Requests performed by the actor $a$ to modify norm $n$ are fulfilled.

**Retroactive Promulgation of a normative power** $\phi = \langle a, n \rangle$: Add $\phi$ to the set of normative powers $P_N$. $P_N = P_N \cup \phi$. Requests performed by the actor $a$ to modify norm $n$ are fulfilled.

**Abrogation of a normative power** $\phi = \langle a, n \rangle$: Remove $\phi$ to the set of normative powers $P_N$. $P_N = P_N - \phi$. Requests performed by the actor $a$ to modify norm $n$ are blocked.

**Annulment of a normative power** $\phi = \langle a, n \rangle$: Add $\phi$ to the set of normative powers $P_N$. $P_N = P_N - \phi$. Requests performed by the actor $a$ to modify norm $n$ are blocked.

Once we have introduced the formal extensions that allow us to support normative context modifications at run-time, we are ready to introduce the monitoring architecture we have designed for effectively supporting monitoring of dynamic normative contexts.

## 3.6 Monitoring architecture specification

This section introduces the architecture of the normative monitor presented in this document, depicted in *Figure* 3.10. Dotted lines represent information flow. Continuous lines represent interfaces. Both types of lines contain arrows indicating the direction of the interaction. The architecture is composed by three main levels:

- The monitoring system, which tracks agent behaviour compliance in terms of organisational norms.
- The agent organisation, composed by the agents operating within the organisation boundaries.
- The norm definition, which contains the high level definition of the norms governing the system and interfaces to updated them.

The norm definition is contained on an *Opera* file created via the *ALIVE* framework [APV+10]. Our system will typically contain an initial norm definition file provided by the system designer. The ALIVE framework provides graphical means to easily edit norm definition files in *Opera* format. The monitor starts by asking the *parser* component to obtain the normative context contained on an *Opera* file.

The parsing process translates the norms inside the normative context to a set of rules that will effectively allow the *rules engine* component to instantiate norms and modify norm instance states. The *rules engine* updates information on the *knowledge base*. As seen in *Figure* 3.10 the monitor's *knowledge base* contains several sets, including:

- IS: With norm instances in instantiated state. That is, instances that have fulfilled their activation condition and are not in transition.
- FS: With norm instances in fulfilled state. That is, instances that have fulfilled their deactivation condition and have not been violated.

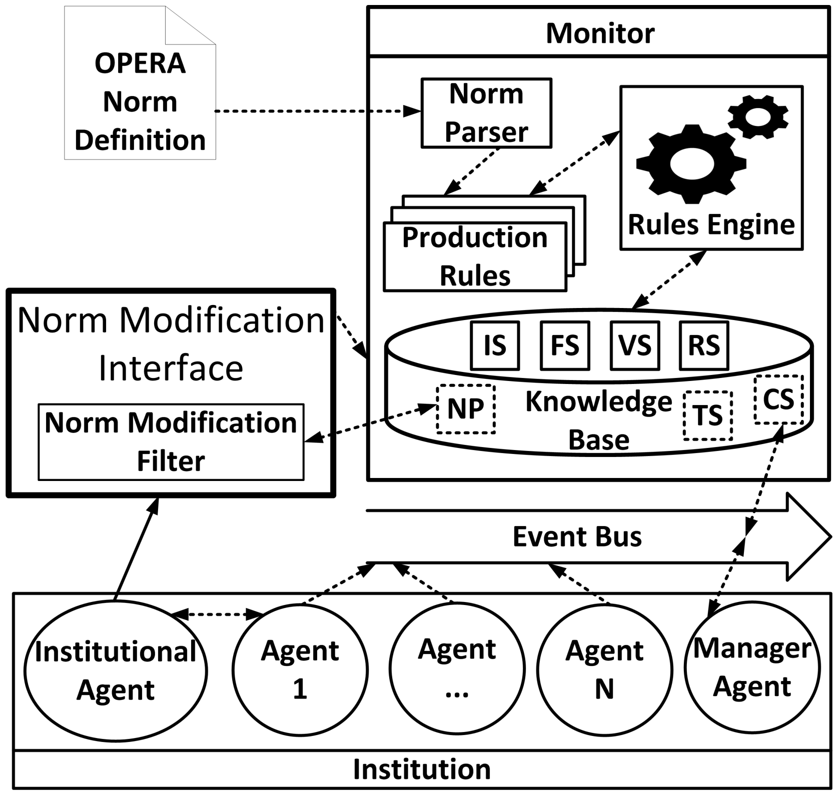



Figure 3.10: Architecture of the normative monitor

- VS: With norm instances in violated state. That is, instances that have been violated. Typically norm instances in this state will trigger violating handling norm instances on institutional agents (*e.g.*, manager agents) so the sanctions associated to the violation can be applied.
- RS: With norm instances in repaired state. That is, norm instances that have been violated and repaired, their sanctions have been applied and their repair actions executed.
- TS: With regulative norms in transition. Associated to norms that have been abrogated and can not be removed from the system yet according to *algorithm* 4.
- CS: With norm instances to be compensated. That is, instances that have been repaired and can require a compensation after the annulment of the associated norm according to *algorithm* 3.
- NP: With the list of institutional powers. It contains both the agents with permissions to update the normative context (set $N_P$ and the list of constitutive powers (set $N_C$) as seen in *definition* 15. The *Opera* file will typically contain an initial definition of normative powers, so the set can be initialized.

The *rules engine* runs a process, that along with the constitutive rules and the constitutive powers, allows for the monitor to infer new events. For instance *in-*

*stanceOf(FrankfurtBar,BarEtto)* can infer *instanceOf(Bar,BarEtto)* and *paid(fine,BarEtto)* can infer *NormInstanceRepaired(SmokingBan,BarEtto)*.

The *event bus* is the component that provides information to the monitor and all the agents, allowing it to perceive the environment. Agent actions that have an impact on the environment are reported to the event bus as *events*. Furthermore, the *event bus* is a bidirectional communication mechanism, that allows the monitor to share inferred knowledge with the set of agents participating in the institution in case it is required. In this aspect, the event bus is used to announce norm violation and norm compensation to manager agents. Therefore, the event bus is the central component to build a shared model of the normative state of the organisation.

Information passed to the agents by the event bus is captured and processed by the *manager agent*. The *manager agent* captures norm instance violations that the *monitor* has inferred and applies the corresponding sanctions and repair actions. Once the corresponding sanctions and repair actions have taken effect, the *manager agent* notifies this by sending a new *event* to the *event bus*. The *monitor* will update the state of the norm instance, marking it as repaired. The *manager agent* must be capable of de-enacting repair actions in case of norm annulment. In this case, the *manager agent* captures norms compensations, applying them by compensating sanctions (*e.g.,* return a fine) and de-enacting repair actions (*e.g.,* take out the ban of an actor).

Finally, the *norm modification interface* allow agents to modify the normative context, adding, removing and updating the norms governing the system. Norms can be updated either by human actors or computational actors. The *norm modification interface* will query the set of normative powers into the *knowledge base* to see if the actor has permissions to update the normative context. In case the actor lacks permissions, the update action will have no institutional effect, and the normative context will remain unchanged. According to *Figure* 3.10 a institutional agent will be responsible of updating the norms in the normative context. In case any other agent (*e.g.,* Agent1 in the figure) wants to change the norm, he will typically send a request to the institutional agent. In case the norm modification request is successful, the *norm modification interface* will run the algorithms introduced in §3.5. For supporting the norm modification algorithms the *norm modification interface* has direct access to the monitor.

It is worth to remark that all norm modification requests pass through the *norm modification interface* component. Therefore, in order to extend normative power support to retroactive operations (please, remember we decided to implement only prospective ones) all we have to do is keep a memory of past requests on the *norm modification interface*. When a retroactive operation is received we can check the memory of the *norm modification interface* component and apply requests retroactively.

Please note that we do not provide support for norm update via modifications of the *Opera* file. That is because the Opera model lacks the concepts of prospective and retroactive norm promulgation and derogation. Therefore, even though the Opera model is the perfect option to support our normative concepts, it is not expressive enough to support our norm modification operations. We could extend the model to support our norm modification operations. However, we consider, in the scope of this PhD thesis, supporting the operations via the *norm modification interface* is enough. We can leave the extension of the Opera model as future work.

Once we have outlined the design of our monitoring architecture for dynamic normative contexts, we can proceed with implementation details.

## 3.7 Implementation

The abstract architecture specification introduced in the previous section is instantiated by the following technological stack:

- **Opera Norm Definition**: Created using the last version of the ALIVE metamodel [2].

```
 {:name "TestWater",
:norms
[],
:cas-rules
[{:context "Universal",
  :concrete-fact
  {:type "predicate",
       :name "ConcrecteTreatment",
       :arguments
       [{:type "constant", :value "Treatment1"}]},
  :abstract-fact
  {:type "predicate",
       :name "AbstractTreatment",
       :arguments
       [{:type "constant", :value "StringentTreatment"}]}
  }]
 }
```

Figure 3.11: Example of parsed norm

- **Monitor Norm parser**: We have used the wire project [3] which contains the function *eu.superhub.wp4.monitor.core.regulative-parser.parse-file* to transform a Opera norm specifications into a Clojure [4] data structure. The data structure in particular is an associative map which can be easily processed. *Figure* 3.11 shows an example of a parsed norm.

```
(eval '(defrule inject-instantiation
        "injecting instantiated norm"
        [?i <- wire.preds.NormInstanceInjected (= ?n norm)
                                               (= ?theta substitution)]
        [:not [wire.preds.Instantiated (= ?n norm) (= ?theta substitution)]]
        [:not [wire.preds.Repair (= ?n2 norm) (= ?n repair-norm)]]
        =>
        (do
          (println "injecting instantiated norm"
          (insert-unconditional! (->Instantiated ?n ?theta))))))
```

Figure 3.12: Example of Clara rule

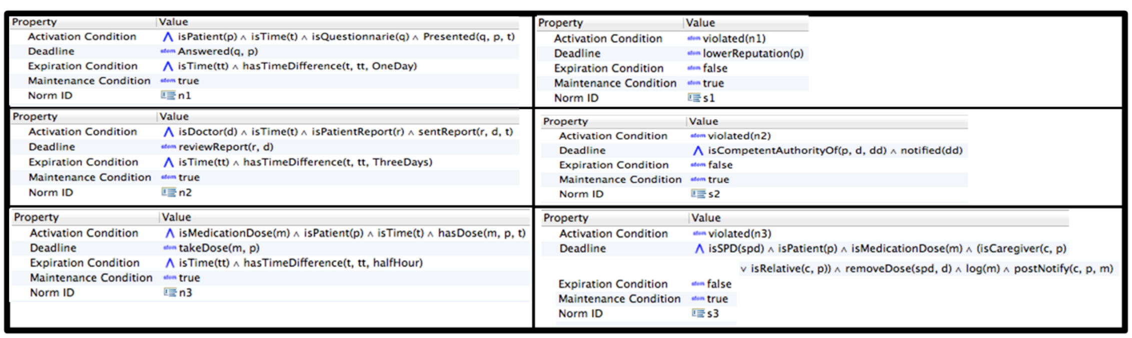
- **Monitor Production rules**: Parsed norms are transformed into Clara rules [5]. Clara is a rules engine with support for clojure. Clara works with a set of production rules, a **Knowledge base** where facts are inserted and retreated, and a **rules engine**. Production rules are composed by two mains sets. The set of parsed norms, that will vary between scenarios and can be updated dynamically and a set of base production rules that are static and constitute the monitor's core. Base production rules are programatically introduced to support the different operations provided by our normative monitor. For instance, *Figure* 3.12 depicts the rule required for injecting active

[2] http://sourceforge.net/projects/ict-alive/
[3] https://github.com/tranchis/wire
[4] http://clojure.org/
[5] https://github.com/rbrush/clara-rules

instances of norms into the knowledge base, which is required for the prospective promulgation operation. It is important to note how the rule is updating the knowledge base in line 10 by inserting an instance of a norm.

```
(defn test-propespective-promulgation-water
  "Test for the propespective promulgation operation using wastewater norms"
  []
   (let [;Take base rules, initial norm set and norm to promulg.
         br (engine/base-rules)
         norm-set-1 generate-norm-set-wow-1
         norm-set-2 generate-norm-set-wow-2

        ;Create Main monitor with initial norm set and inject event
        monitor-1 (create-monitor norm-set-1 br)
        monitor-1 (update-monitor monitor-1 {:name "NumberOfWorkers"
                                             :argument-0 "x"})
        monitor-1 (update-monitor monitor-1 {:name "lessThan"
                                             :argument-0 "x"
                                             :argument-1 "5"})
        monitor-1 (apply-update monitor-1)

        ;Prospectively promulgate a norm
        _ (info "propespective promulgation of a new norm")
        monitor-1 (propespective-promulgation monitor-1 norm-set-2 br)

        ;Inject a new event to verify norm is operative after merge
        monitor-1 (update-monitor monitor-1 {:name "Unit"
                                             :argument-0 "z"})
        monitor-1 (update-monitor monitor-1 {:name "Unit"
                                             :argument-0 "z"
                                             :argument-1 "Archmage"})
        monitor-1 (apply-update monitor-1)

        ;This is the result
        test-instantiated (get-instances-in-state monitor-1 "Instantiated" br)
        #_ (pprint monitor-1)
        result test-instantiated ]
    result))
```

Figure 3.13: Example of parsed norm



| Property | Value |
|---|---|
| Activation Condition | ⋀ isPatient(p) ∧ isTime(t) ∧ isQuestionnarie(q) ∧ Presented(q, p, t) |
| Deadline | Answered(q, p) |
| Expiration Condition | ⋀ isTime(tt) ∧ hasTimeDifference(t, tt, OneDay) |
| Maintenance Condition | true |
| Norm ID | n1 |

| Property | Value |
|---|---|
| Activation Condition | violated(n1) |
| Deadline | lowerReputation(p) |
| Expiration Condition | false |
| Maintenance Condition | true |
| Norm ID | s1 |

| Property | Value |
|---|---|
| Activation Condition | ⋀ isDoctor(d) ∧ isTime(t) ∧ isPatientReport(r) ∧ sentReport(r, d, t) |
| Deadline | reviewReport(r, d) |
| Expiration Condition | ⋀ isTime(tt) ∧ hasTimeDifference(t, tt, ThreeDays) |
| Maintenance Condition | true |
| Norm ID | n2 |

| Property | Value |
|---|---|
| Activation Condition | violated(n2) |
| Deadline | ⋀ isCompetentAuthorityOf(p, d, dd) ∧ notified(dd) |
| Expiration Condition | false |
| Maintenance Condition | true |
| Norm ID | s2 |

| Property | Value |
|---|---|
| Activation Condition | ⋀ isMedicationDose(m) ∧ isPatient(p) ∧ isTime(t) ∧ hasDose(m, p, t) |
| Deadline | takeDose(m, p) |
| Expiration Condition | ⋀ isTime(tt) ∧ hasTimeDifference(t, tt, halfHour) |
| Maintenance Condition | true |
| Norm ID | n3 |

| Property | Value |
|---|---|
| Activation Condition | violated(n3) |
| Deadline | ⋀ isSPD(spd) ∧ isPatient(p) ∧ isMedicationDose(m) ∧ (isCaregiver(c, p) ∨ isRelative(c, p)) ∧ removeDose(spd, d) ∧ log(m) ∧ postNotify(c, p, m) |
| Expiration Condition | false |
| Maintenance Condition | true |
| Norm ID | s3 |

Figure 3.14: Example of ALIVE norm

- **Norm modification interface**: As NoMoDEI is currently a prototype we are actually retrieving norms from a MongoDB database [6] and injecting them on the system using live testing functions, adapted to our different scenarios. Therefore, the interface,

[6] `https://www.mongodb.org/`

along with the **Norm modification filter** is not currently supported. However, as we are already using a norm model decoupled from the monitor component, it would be very easy to integrate the interface with the database. *Figure* 3.13 depicts one of our testing procedures for Prospective Promulgation.

- **Event Bus**: The event bus is based on RabbitMq technology [7] where messages are observed from the agents and passed to the monitor. As seen in *Figure* 3.13 the messages are captured, along with their arguments, and inserted into the monitor.
- **Agents**: Messages for the prototype are directly generated via testing functions. On the tests based on our example scenarios, autonomous agents generate the messages and provide them to the monitor.

As a summary, NoMoDEI makes use of the following technologies:

- ALIVE framework: Provides a metamodel of norms which is expressive and complete enough for the scope of this thesis. Includes both constitutive and regulative norms based on deontic logics. Norms are connected to an organizational model including roles and partial states descriptions of the world and to an OWL ontology so both organizational and ontological concepts can be effectively referenced in the norms. The metamodel lacks the concept of constitutive power and normative power, as well as a model of the different operations for norm dynamics. We plan to extend the metamodel to cover such concepts as future work. *Figure* 3.14 shows an example of several ALIVE norms related to one of our test scenarios. The reason to select ALIVE instead of another alternatives (such as RuleML [Gov05] or SWRL [HPSB$^+$04]) is that it is expressive enough and easy to use. Furthermore we have already some experience using the ALIVE metamodel.
- Wire project: Provides a basic non-dynamic norm monitoring framework as well as several auxiliary functions (*e.g.*, parsing norm definitions specified using the ALIVE metamodel). The project has been selected because it provides a basic norm monitoring framework that is easy to understand and extend due to the tidy and well-written code structure. The project lacks deeper documentation but support was provided eagerly by the main developer on demand.
- Clojure: We have decided to use Clojure for extending the wire project. The original wire project is already written in Clojure, and porting it to another programming language would incur in a development cost. Furthermore, the Clojure programming language presents some useful properties, such as:
  - Runs on top of a Java Virtual Machine, uses Java code for integrating with other components and Java interop for transparently using Java libraries and projects.
  - Facilitates working with maps, vectors and specifically data encoded in JSON format.
  - Homionicity and first class functions. Supports evaluating data as code in case it is required.
- Clara rules: We have selected this technology over other alternatives (*e.g.,* Jess [8] and Drools [9]) mainly because it is easy to integrate with Clojure code, provides an immutable consistent working memory and most data structures can be treated as Clojure data structures (*e.g.,* rules and facts) and therefore, easily processed.

[7] `https://www.rabbitmq.com/`
[8] `http://herzberg.ca.sandia.gov/`
[9] `https://en.wikipedia.org/wiki/Drools`

- Rabbit mq: We have selected this technology because it is a robust cross-platform messaging system that provides slight performance gains over similar systems, such as JBoss [10].
- MongoDB: MongoDB is a document store supporting JSON format. Document stores are used for storing semi-structured document object data and metadata. Documents can be queried by their properties in a similar manner to relational databases but are not required to adhere to the strict structure of a database table, like in typical relational databases. We have selected MongoDB because it does not require the design of a tight data model and integrates seamlessly with Clojure data structures.

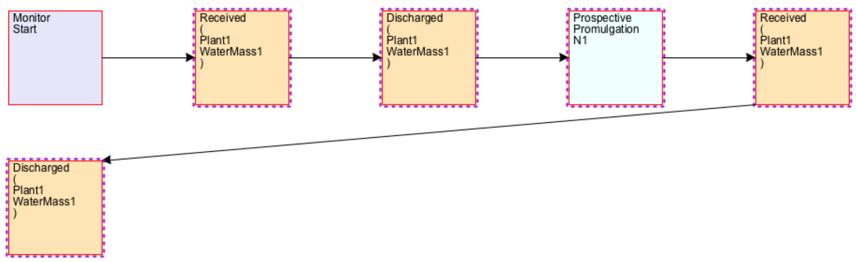



Figure 3.15: Example of visualization time-line

It is important to note we have extended our base architecture during implementation including a norm visualization component. When inferring relevant events (*e.g.*, norm instantiations, violations *etc.*) the monitor populates a visualization database that is periodically queried and visualized. The visualization is inspired by state of the art norm visualization techniques [COML12] and provides live information on the normative state of the system. The visualization system uses MongoDB for data storage, is implemented in Clojure and is currently using the lacij library [11] to generate the visualizations. Two main visualizations are available, time-line visualization and norm visualization. On the one hand, time-line visualization, as depicted in *Figure* 3.15 represents the stream of events received and generated by the monitor. On the other hand, a norm visualization, as seen in *Figure* 3.16, depicts the set of events relevant for the different states of a norm and colours them depending on weather they have already ocurred (green box) or have not happened yet (red box). It is important to note the norm visualization is able to cope with logic operands such as negation and disjunction. *Chapters* 4 and 5 provide more information on norm visualizations applied to our tests scenarios.

The NoMoDEI software is currently available as a fork of the wire project, which has been extended for supporting dynamic normative monitoring. The project is hosted at [12]. The project can be downloaded, installed and tested following the instructions provided on the *README.md* file. The project is open source (GPL license) so the source code is also hosted at the mentioned URL. The norm visualization component is deployed in a private repository [13]. Access to the repository can be provided on demand.

[10] http://docs.jboss.org/jbossmessaging/docs/guide-1.0.1.SP5/html_single/
[11] https://github.com/pallix/lacij
[12] https://github.com/ignasi-gomez/wire
[13] http://kemlg.mooo.com/igomez/nomodei/tree/master

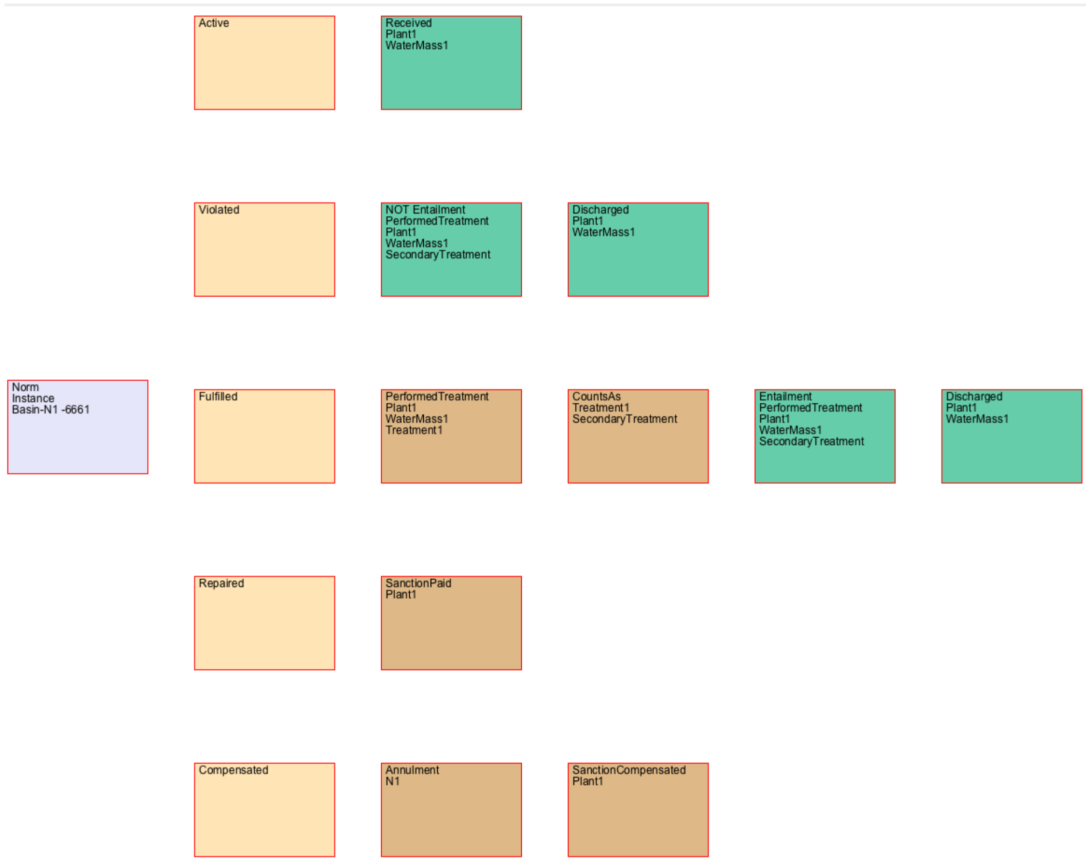



Figure 3.16: Example of visualization norm

Once we have introduced our implementation details, we can proceed with conclusions.

## 3.8 Discussion

The NoMoDEI framework introduced in this chapter provides support for monitoring dynamic normative contexts that can expand and contract. Expansion and contraction operations are performed on the fly, without having to stop the monitoring process. In this section we have presented the four normative context operations we implement in order to support norm dynamics. We have outlined the formal extension of the base monitoring model, required for implementing these operations and algorithms for performing them. Then, a monitoring architecture for supporting dynamic normative contexts is presented. Finally, implementation details will enhance this chapter in the final version of this document. Our proposed framework will effectively provide support for monitoring scenarios with dynamic normative contexts, and what is more important, scenarios with multiple normative contexts where the normative monitor has to jump from one context to another at run-time.

There is currently an important amount of work being done in context change management:

- Governatori [GR08b] proposes an extension of his logics for normative monitoring that enables capturing the different temporal aspects of *abrogation* and *annulment*. The extension increases the expressive power of his logics allowing it to represent meta-norms describing norm modifications. Meta-norms refer to a variety of possible time-lines through which conclusions, rules and derivations can persist over time. In particular, the extension defines temporal constraints that permit either allowing for or blocking persistency with respect to specific time lines. The idea behind Governatori's approach is blocking of derivations across repositories (*i.e.*, timelines). When a modification is applied to the normative context, it is split into two repositories: where the modification occurs and where it does not. For instance, if a norm is *abrogated*, norm's conclusions are derived only in the repositories where the rule has not been *abrogated*. When compared to our approach, Governatori's has the following drawbacks:
    1. Norm *annulment* presents a problem under this approach, conclusions of *annulled* norms might remain on the repository after the norm has been *annulled*, and the solution proposed to remove the conclusion seems quite ad-hoc.
    2. Governatori's solution provides no explicit support for *retroactive promulgation*.
    3. Governatori's approach is not able to update the deontic part of the context (*i.e.*, obligations and permissions), in fact Governatori states that an explicit differentiation between norms, obligations and permissions has to be made.
    4. Governatori's approach does not provide support for *classificatory rules* (*i.e.*, counts-as).
- Aucher's proposal [AGHL09] is mainly theoretical. This approach is similar to ours, in the sense both are event-based, it does not make distinctions between deontic statements and norms and has full support for *classificatory rules*. However, neither context expansion nor contraction provides support for *ex tunc* operations. In fact, only one expansion and one contraction operations are introduced on his approach; both operations seem to (implicitly) be of *ex nunc* type. Means for ensuring that the normative context is consistent (after the expansion/contraction operation) are included on Aucher's approach, whereas we have not taken care of such issues. As seen in §3.6 we use the ALIVE framework [APV+10] for specifying our norms. As ALIVE relies in the Opera methodology [Dig04] it ensures modifications result in a consistent and non-redundant model via model checking techniques. Therefore we can assume the normative context we deal with is consistent and non-redundant. Please notice that non-redundancy is an interesting property on normative contexts (can increase performance when reasoning about the model), however, it is neglected on Aucher's approach.
- Campos' approach [CLSRAE09] raises from the need of turning *Electronic Institutions* (EI from now on) into *Situated Electronic Institutions* (SEI from now on). *EI* are static and self-contained, agent actions are filtered so norm violations will never occur. *SEI* control over external agents is not tight, therefore violations can occur, and *SERI* can adapt themselves to changes in the dynamic existing social systems. Campos' approach uses a *Bridge* for communicating the *SEI* with the environment. The *Bridge* is

similar to the *event-bus* we use on our approach, but contains an API tightly coupled with the domain the environment refers to, while our *event-bus* is domain independent. Besides this, Campos' approach does not support *classificatory rules*. However, Campos' approach allows for *SEI* to automatically adapt the normative context in order to perform better in new environments. Re-configuration is achieved via *transition functions* (TF) that define some basic updates on the normative context of the *SEI*. For instance, if violating the norm $N$ implies the payment of a fine, and the system detects the number of violations of $N$ is higher than the expected value, a TF can increase the value of the fine. Thus, TF define very simple modifications to the normative context, without support for norm promulgation, abrogation and annulment.

As a summary, the main contributions of this chapter are:

1. A conceptualization of a norm monitoring framework supporting the context update operations.
   a) We include four operations supporting prospective and retroactive norm insertion and deletion. We also define norm updates.
   b) The operations can be applied to regulative and constitutive norms.
   c) The conceptualization is enhanced to support constitutive powers and normative powers.
2. A formal model of the norm monitoring framework, including:
   a) A formal model of the four norm update operations.
   b) Formal algorithms for supporting the operations.
   c) Extension of the formalism, so the operations can support both regulative and constitutive norms. The extension is also supported via formal algorithms.
   d) Extension of the formalism to include constitutive powers and normative powers. The extension is also supported via formal algorithms.
3. A base architecture for the norm monitoring framework.
4. An extension of the base architecture to support norm update operations, constitutive powers and normative powers.
5. An implementation of the architecture.
6. A prototype for a norm visualization component for documenting the results of our research.

We have based our work in these previous existing works:

1. The norm model we use is directly based on the ALIVE metamodel for norms [APV+10], so it is not an original contribution. We have collaborated in defining this metamodel in the past.
2. The basic norm monitoring framework and architecture we use is directly based on the norm monitoring framework defined by Alvarez-Napagao *et al.* [AÁNDVS10], so it is not an original contribution. We have collaborated in defining the architecture in the past. We have extended the original framework and architecture during this chapter to support norm dynamics, normative powers, constitutive powers and a high level visualization architecture.

In this chapter we started by providing the formal details of our proposed framework. We have introduced a base framework and extended it to support norm dynamics. We have also introduced the formalisation including both architectural and implementation details. In general, this chapter can be considered as a complete description of NoMoDEI, our proposed framework. In chapters §4 and §5 we apply NoMoDEI to demonstration scenarios in order to show the utility and applicability of our framework to real world problems. Specially problems that can be tackled using information technologies in general and Multi-Agent Systems in particular. §4 applies NoMoDEI to wastewater management on a river basin. §5 applies NoMoDEI to Ambient Assisted Living for elder patients.

CHAPTER 4

# A Practical Use case: Wastewater management on the Besos River

Information technologies applied to environmental issues show potential in a wide range of fields, among others, decision support systems [PCRR$^+$04] and simulations [H$^+$69]. However, the complexity of environmental problems introduces several challenges that information technologies should tackle. The first one is the fact that environmental issues must be considered in terms of complex systems, mainly due to the amount of variables to be considered and their dependencies. Also the high degree of uncertainty associated to the system and the potential impact (and therefore, risk) of the decisions taken *w.r.t.* these systems. The second issue is the fact that, in environmental systems, the scenario should often reflect conflicting goals, and we need to take into account a set of heterogeneous (sometimes conflicting) views and perspectives. In the particular case of Wastewater management a variety of factors (*e.g.*, economical, technical, ecological, *etc.*) are to be considered, and associated with each factor is a different set of goals. These complex scenarios, where a wide variety of actors with different (sometimes conflicting) goals interact between them, can benefit from norm-aware electronic distributed systems based on agent technologies. Such systems can ensure compliance with the different actors to the expected behaviours and environmental policies, where environmental policies are designed to guide the overall system to a common higher goal, such as the preservation of the environment while keeping an active economy.

In this *Chapter* we propose a norm-aware agent-based model for integrated Wastewater management systems. The idea of using Autonomous Agents to cope with the problem has been done in view of the various, sometimes conflicting, goals that the identified actors have to fulfil their private interests. In this scenario each actor requires its own system *view* with customized privileges and access to differing control tools, either managerial or operational.

The agent-based model is applied to the scenario depicted in *Figure* 4.1. The system proposed aims at managing the wastewater treatment capabilities of WWTPs, coordinat-

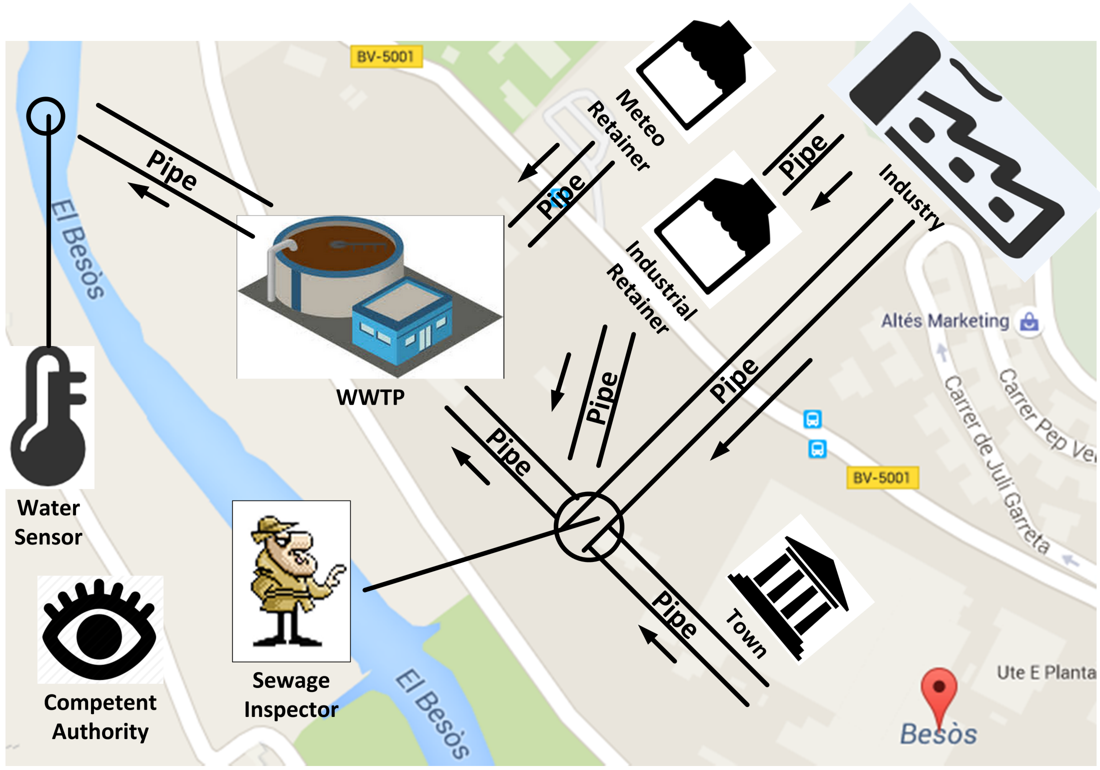



Figure 4.1: Sample wastewater management scenario

ing them with another plants and the different actors in the scenario. Plants treat wastewater coming from different sources before discharging it to the river, treated and with the appropriate ecological conditions. To ensure river's ecological quality, water sensors measuring different parameters (*e.g.*, temperature, acidity, suspended solids on water, river oxygen demand, *etc.*) are located along the river in interesting points, such as water discharge points for WWTPs. Plants are connected to different elements providing wastewater with different characteristics, including: towns providing househould wastewater with a steady flow of quantity and variable pollutant concentrations; rain retaining tanks providing lightly polluted meteo wastewater, which comes in very high quantities during short periods of time; industries providing industrial wastewater with high variability both in quantity and pollutant concentrations. Some of the elements (*e.g.*, towns) are connected directly to the plant, effectively providing a steady flow of wastewater. Some elements have a retainer tank between the wastewater source and the plant (*e.g.*, metereological wastewater).

Also, some elements present both options, they can discharge wastewater directly to the plant or store it on a retention tank (*e.g.*, industries). The different elements are connected using pipes that form a sewage network. Some points in the network (*e.g.*, where industrial wastewater is mixed with househould wastewater) might be observed by sewage inspectors. Finally, a competent authority oversees the whole system taking decisions to ensure both the ecological quality of the water and the economic sustainability of the society.

The idea behind applying an agent-based model to the scenario is supporting social simulations and the study the impact of social norms [Axe86] on a river basin. In social simulations human reasoning activities are emulated via computational methods. When embedding social constructs such as norms and social structures into agent based simulations we are effectively bridging the gap between simulations and social simulations. Therefore, this *Chapter* provides an extensive description of the social structures used by our agent-based model. Specifically, our Agent-based model is supported by the *ALIVE* [APV+10] framework. This *Chapter* focuses on the specification of *ALIVE* 's organisational model, putting special emphasis on the norms and how they evolve due to organisational, technological, social and contextual changes.

The rest of this Chapter is organized as follows. Section §4.1 provides an introduction to the problem of integrated management of wastewater systems, analysing previous approaches and outlining our proposal to tackle this problem. Section §4.2 describes the process followed, and the case study. It is an urban wastewater system inspired on the actual Besós river basin which is fully described in §4.2.1. Section §4.3 explores the objectives, roles and social structure of the system with the communication links. In §4.4 we introduce the basis and elements for the decision making from a wastewater management perspective. Later, §4.5 introduces the norms governing the system and provides examples of how they can evolve dynamically. Later, in §4.6, we document the implementation of our test framework focusing on the architecture and the norms resulting from the tests in our scenario. Finally, §4.7, presents a discussion and the main conclusions of this chapter.

## 4.1 Integrated management of wastewater systems

The integrated management of wastewater systems has acquired notorious importance since the beginning of the $21^{st}$ century. The combined pollution considered in this approach establishes the relation between the characteristics of effluent from urban wastewater systems (UWSs) and the quality of the receiving waters. In order to obtain an effluent with pollution levels according to the values that preserve an adequate state of receiving waters, the processes of all stakeholders should be performed in a coordinated way. But it becomes a difficult management task due to the intertwined factors and the different (macro, micro) scales of the processes involved.

In literature, several authors (*e.g.*, [Van96], [RAK+98], [BS05]) propose different approaches focused on the receiving waters' objectives for solving the UWSs integrated management. In [CAGP13] the need for integrated wastewater management policies is analysed, with the aim of allowing a more efficient and sustainable management of wastewater treatment plants, effectively maximizing the ecological, economical and social benefits of the system as a whole.

Rauch [RAK+98] proposes to simplify the integrated model focusing on a significant impact of wastewater discharges although he describes difficulties to combine the models. He proposes avoiding compatibility problems through the definition of the relevant impact on receiving waters, the identification of the minimum set of state variables, and the processes necessary for defining the impact, in addition to the specification of inputs and transformations in the model interfaces. Butler [BS05] focuses his research by considering the performance criteria through a tool that includes simulation models for the components. Seggelke [SRVK05] studies the control of the wastewater treatment plant (WWTP) influent as a factor that contributes to reduce the global impact on the receiving waters. Schmitt [SH06] evaluates the *permissiveness* to exclude system components or interactions. He remarks that the system complexity and heterogeneity. Meirlaen [MHS+01] proposes the use of substitution models for reducing the complexity. According to this study, substitution models provide lower (but sufficient) accuracy while reducing computational complexity, effectively reducing execution times. However, it requires an increased amount of data for calibration purposes. A further work of the [MVAV02] proposes the real-time control as a valid option for minimizing the impact of effluent on receiving waters, in similar way to the proposal of the revision work by Schutze [SCC+03]. Furthermore, for significant interactions, Erbe [ERSL02] exposes the requirement of simultaneous numerical simulations of discharges in the sewer system and the treatment. In later works [EFG+02, ES05] he proposes flow analysis through the overall system in order to manage in terms of treatment state variables or of receiving waters. Benedetti [BBB+05] studied the identification of pollutant flows according to the supply system.

In order to manage the complexity of wastewater systems, Rendón-Sallard [RSSMAC06], Poch [PCRR+04], Cortés [CSMC+00], Comas [CPRRC02], Makropoulos [MNL+08] and Benedetti [BPN+09] propose the use of decision support systems. In this context, the introduction of autonomous decision support systems, defined as *agents* (Wooldridge [Woo97]), has opened a new way in management systems (Cortés and Poch [CP09]). This approach allows modeling the interactions between components and/or stakeholders and to model the different variable scales of the elements in the domain. Agents have capacity to acquire local environmental data through sensors and to act over their surrounding environment in real time without the need of a centralized controller. They have both individual and collective aims that drive their behavior. The former ones typically consist of both environmental and economic goals from individual stakeholders, which can be achieved individually with the agents' own resources and capabilities. The latter ones typically refer to societal aims and challenges and require the participation of several agents in order to be achieved (Sycara [Syc98]). This set of agents (multiagent system, MAS) show a collective behavior to achieve the aims inaccessible by individuals. Mataric [Mat95] considers the collective behavior of agents as the temporal pattern of interactions and their direct or indirect communications as the more habitual way to interact. The action limits of agents [Bar99] would be managed in a negotiated and coordinated way [Jen01, Cas98]. Therefore one central aspects of multiagent systems is to coordinate actions. *Coordination* is the process of managing dependencies between activities [MC94] By such process an agent reasons about its local actions and the foreseen actions that other agents may perform, with the aim to make the community to behave in a coherent manner. In the context of multiagent systems, coordination mechanisms are necessary for the effective operation of all stakeholders in order to get a well-balanced division of labour while reducing logical coupling and resource dependencies of agents.

Literature shows different methodologies for developing systems of agents. Several widely referenced are Gaia [WJK00b, ZJW03], MaSE [DWS01, WD01], Tropos [BPG+04] and Prometheus [PW03]. The analysis of these methodologies shows that all (with the exception of Prometheus) present difficulties for considering different abstraction levels [SS04]. However, as Zambonelli remarks in [ZO04], generic MAS methodologies pose some difficulties when developing real, complex applications. Indeed, the performance characteristics of wastewater systems, with characteristics of influent, effluent, and receiving waters changing, and new wastewater inputs entering to the system with wastewater being retained and treated by infrastructures from different stakeholders can be partially modeled as the result of the interaction of different actors.

However, modern wastewater systems are the result of the interaction of several public and private stakeholders with clear responsibilities and complex interdependencies. We require a more expressive framework to model not only the actors' capabilities, actions and effects in the system, but also some social factors such as their individual and collective responsibility, the ways interactions are structured in agreed patterns, social phenomenon such as authority, power and influence, and the existence of regional, national and international regulations.

With all this in mind we have selected the *ALIVE* Framework [VSVP+10]. *ALIVE* is both an agent-oriented software modeling framework and a methodology for the design distributed computational systems. It is the result of the FP7 European project *ALIVE*. The framework allows designing and implementing systems, taking into account organisational, coordination and service perspectives [ÁNCPVS09]. Basically, *ALIVE* can be applied in highly regulated scenarios where changes can occur at either abstract or concrete levels, and where services are expected to be continuously changing, with new services entering the system and existing services leaving it at run-time [DDPVS09]. *ALIVE* has been applied both for the dynamic orchestration of distributed, on-line services and even to support dynamic, flexible coordination of human teams in crisis management scenarios [DDPVS09].

We propose that each actor is represented by an agent, see §4.3. Modeling and simulating of such an integrated system is a solid means to transpose our actual knowledge of that system into predictions of it and will bridge the existing gap between theory and practice (see [RSSMAC06]) The system has been developed using state-of-the-art knowledge about the Besós river basin.

This *Chapter* presents the high-level organizational structure for the integrated management of an urban wastewater system conceptualized as a MAS. The new proposal is developed based on the *ALIVE* methodology. The approach focuses not on the configuration of interactions between the main functions of the system in the environment, but on the (individual and collective) responsibilities of every stakeholder and how they depend on other in order to achieve their (individual and collective) goals, ensuring an adequate global system performance. This target means that the characteristics of wastewater effluent would be as adequate as possible for the expected quality in receiving waters.

## 4.2 METHODOLOGY

In the organizational model described in *ALIVE* roles are the central concept. Roles identify the activities necessary to achieve organizational objectives and enable abstraction from the specific actors that perform them [DDPVS09]. Based on these conceptualizations, the modeling process follows an iterative application of the following steps:

- identify the stakeholders in the system
- formally define the roles, identifying their goals and their dependencies
- model the interaction scenes between roles in order to manage every single dependency

- organize the scenes into a coherent interaction structure
- identify the way agents will enact roles at run-time.

All this process is supported by the OperettA Tool [AD11], this is one of the results of the *ALIVE* FP7 project [VSVP$^+$10].

### 4.2.1 Schema of wastewater flows in the case study

The case study is a River Basin composed by elements generating both wastewater (a set of households $[\kappa_1, \ldots \kappa_k] \in \mathrm{K}$ and a set of industries $[I_1, \ldots I_i] \in \mathcal{I}$) and polluted water (meteorological events that generate runoff). For simplicity we will consider both wastewater and polluted water to be wastewater. There are also elements storing wastewater (a set of retention tanks $[T_1, \ldots T_l] \in \mathcal{T}$), treating wastewater (a set of Urban Wastewater Treatment Plants $[W_1, \ldots W_j] \in \mathcal{W}WWTP$), and receiving waters (*e.g.* a River).

Also, there is a graph $s_i \in \mathcal{S}$ that represents the sewerage infrastructure in a urban sector or city. It encompasses components such as receiving drains, manholes, pumping stations, storm overflows, and screening chambers of the combined sewer or sanitary sewer. $s_i$ ends at the entry to a $wwtp_j$. In turn every $wwtp_j$ is connected with the receiving waters. In our model as in many European countries all elements in K and $\mathcal{I}$ are obliged to connect their sanitation and/or wastewater discharge to $s_i$ where possible.

The wastewater is characterized by the flow (or volume) and the pollutant concentrations of: Total Suspended Solids ($TSS$), Biochemical Oxygen Demand ($BOD$), Chemical Oxygen Demand ($COD$), Total Nitrogen ($TN$) and Total Phosphorus ($TP$), which are defined as the set of pollutants $x_r$ with $(TSS, DBO, DQO, TN, TP)$ = ($x_1$, $x_2$, $x_3$, $x_4$, $x_5$) [VCP12]. All concentrations related to these pollutants are indicated with a supra-index $r$, with $r$=1, . . ., 5. For subsequent paragraphs, this specification is not repeated in the text in order to avoid many repetitions.

Figure 4.2.1 shows the schema of the case study components, with arrows indicating the wastewater flow. The household generates a wastewater mass $M_D \in \mathcal{M}$ with a particular volume $volume(M_D) = V_D$, and a concentration for each pollutant $O_j \in \mathcal{O}$ $concentration(M_D, O_j) = C_D^j$, which is discharged in a plant $W_k \in \mathcal{W}$. Analogously, the runoffs retention tank has a wastewater mass stored $M_M \in \mathcal{M}$ with volume $volume(M_M) = L_M$ and pollutant concentration $concentration(M_M, O_j) = C_M^j$. The tanks has a volumetric discharge to $W_k$ and feasible volumetric bypass to receiving waters when the retention tank has an overflow. It means the tank can bypass a water mass $M_{DM} \in \mathcal{M}$ with a volume $volume(M_{DM}) = V_{DM}$. These two discharge possibilities allow adapting the sewer performance of separative or combined run-off collection. Each industrial activity has its own retention tank, with a water mass $M_i \in \mathcal{M}$ with volume $volume(M_i) = L_i$ and pollutant concentration $concentration(M_i, O_j) = C_i^j$. Its volumetric discharge to the treatment is $volume(M_i) = V_i$. The plant $W_k$ is capable of accepting a water mass as influent $M_T \in \mathcal{M}$ with volume $volume(M_T) = V_T$ and pollutant concentration $concentration(M_T, O_j) = C_T^j$. It provides a water mass as effluent $M_e \in \mathcal{M}$ with volume $volume(M_e) = V_e$ and pollutant concentration $concentration(M_e, O_j) = C_e^j$ to receiving waters.

Additionally, the treatment has the possibility to bypass wastewater. The bypass consists in a water mass $M_b \in \mathcal{M}$ with volume $volume(M_b) = V_b$ and pollutant concentration $concentration(M_b, O_j) = C_b^j$. The upstream provides a water mass $M_U \in \mathcal{M}$ with volume $volume(M_U = V_U$ and pollutant concentration $concentration(M_U, O_j) = C_U^j$ to receiving waters. The receiving waters correspond to a section of river basin, which has a water mass $M_{RW} \in \mathcal{M}$ with volume $volume(M_{RW}) = V_{RW}$ and pollutant concentration $concentration(M_{RW}, O_j) = C^j_{RW}$.

## 4.3 Wastewater Systems' Organizational Model

Agent based systems [WJ95] are an alternative for designing and implementing open and dynamic systems. As defined by Wooldridge and Jennings: *An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives*. Agents are capable of social behaviour, they can communicate, compete and cooperate among them.

Being capable of social behaviour, the idea of building agent systems which capture notions from human society is recurrent in Artificial Intelligence research. The main idea behind a society is to allow its members to coexist in a shared environment and pursue their respective goals in cooperation or competition with others. Therefore, artificial social systems [MT95] define an abstract social level over computational systems. The social level models the multi-agent system as an organization of entities, defining structured patterns of behaviour that facilitate and enhance the coordination of agent activities [VS03], effectively providing an *Organizational Model* agent's can understand and use.

This section introduces the Organizational Model of our scenario, including the *Social Structure* (with roles and their relationships) the *Interaction Structure* (with the *Landmarks*, patterns of interaction by which agents coordinate their behaviour) and the *Social model* (mapping abstract roles to particular agents).

### 4.3.1 Social Structure

In *ALIVE* the *Social Structure* allows the description of the roles and their relationships, connecting them with both the individual goals and the societal aims. In our proposal the global aim of the wastewater systems' organizational structure is to achieve an effluent with characteristics adequate to the quality requirements of receiving waters. The roles model the distribution of responsibilities among stakeholders and their dependencies. Figure 4.2 shows the set of roles and their associated objectives and sub-objectives.

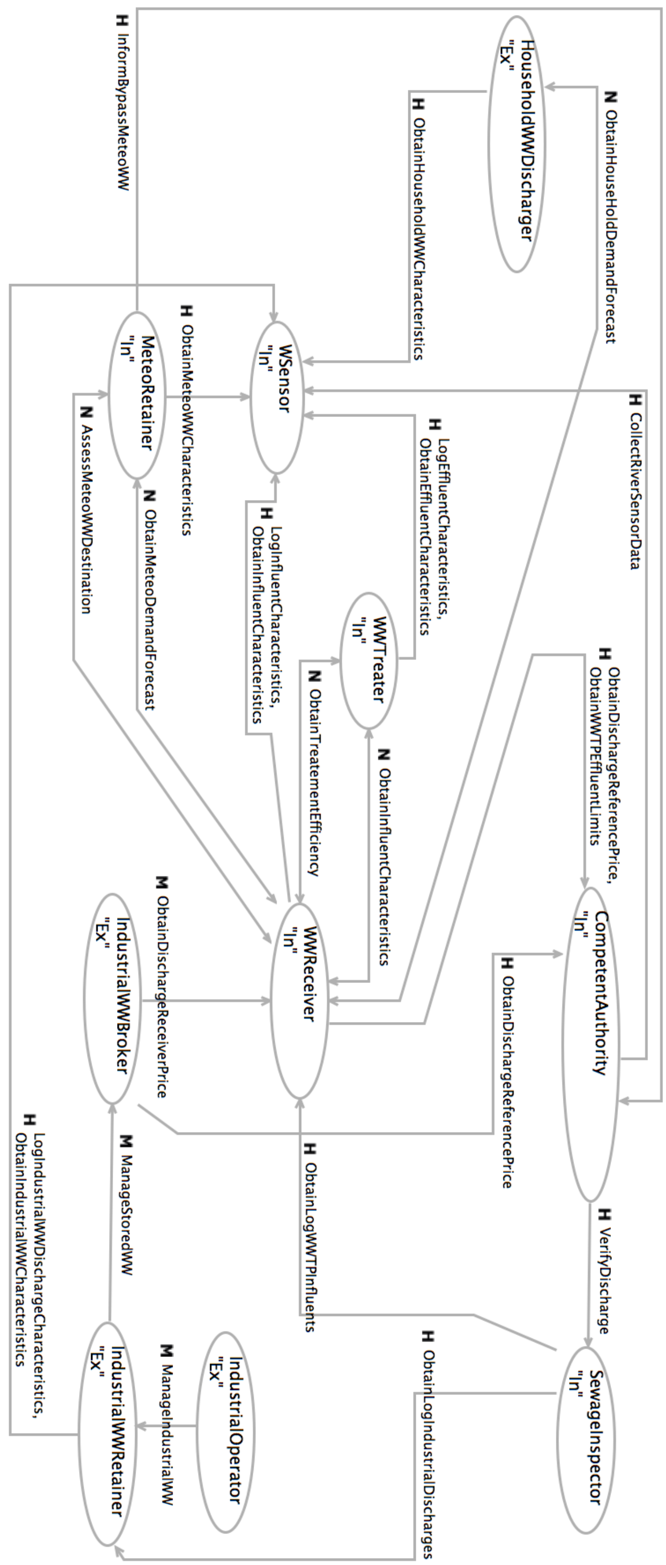



Figure 4.2: General Scene pattern

Table 4.1 provides an enumeration of roles and their objectives and sub-objectives. Some objectives (in *italics*) depend on another role, *via* the role interaction structure, to be achieved. *Figure* 4.2 defines scenes that describe how agents coordinate, and how transitions are done from one scene to another. Role dependencies come in three flavors, *hierarchically dependencies* (where the parent role has some form of authority over the child role, and therefore when the parent role requests the child role to perform a task, the child role is expected to abide), which are indicated with $H$ in Figure 4.2, *network dependencies* (where roles coordinate themselves as peers by mutual interest and support each other to fulfill a common goal) and *market dependencies* (where there is a set of producer roles offering information and/or services to consumer roles for a given price), which are indicated with a $M$ in Figure 4.2. Also in Figure 4.2 External $Ex$ and Internal $In$ roles are shown. Internal roles are roles controlled by the organization. Typically, if they are software components it means the organization has access to the software source code and is able to control and verify it. External roles are roles participating in the organization, but not controlled by it. Following the same example, in the case of software components it means they have not been necessarily developed by the organization, and therefore there it might be no way to access the component's code and formally verify its behaviour.

| **Role** | **Objectives** |
|---|---|
| Industrial Operator | • MakeProfit<br>  – Produce<br>  – ManageIndustrialWW (Depends on *IndustrialWWRetainer*) |
| Industrial WWRetainer | • StoreIndustrialWW<br>• ManageStoredWW (Depends on *IndustrialWWBroker*)<br>  – DischargeIndustrialWW<br>• LogIndustrialWWDischargeCharacteristics (Depends on *WSensor*) |
| Industrial WWBroker | • AssessAmountOfIndustrialWWDischarge<br>  – ObtainDischargeReservedCost<br>    ∗ ObtainDischargeReceiverPrice (Depends on *WWReceiver*)<br>    ∗ ObtainDischargeReferencePrice (Depends on *CompetentAuthority*) |
| Household WWDischarger | • ObtainHouseholdWWCharacteristics (Depends on *WSensor*)<br>• DischargeHouseholdWW |

| | |
|---|---|
| Meteo Retainer | • StoreMeteoWW<br>• ObtainMeteoWWCharacteristics (Depends on *WSensor*)<br>• AssessMeteoWWDestination<br>• DischargeMeteoWW<br>• BypassMeteoWW<br>  – InformBypassMeteoWW (Depends on *CompetentAuthority*) |
| Competent Authority | • EnforceWaterQualityPolicies<br>  – AssessRiverWaterQuality<br>    * CollectRiverSensorData (Depends on *WSensor*)<br>  – InformExceptionalRestrictions<br>• VerifyDischarge (Depends on *SewageInspector*)<br>• VerifyWWTPEffluent<br>  – CollectRiverSensorData (Depends on *WSensor*)<br>• CalculateDischargeReferencePrices<br>• CalculateEffluentLimits<br>  – AssessRiverWaterQuality<br>    * CollectRiverSensorData (Depends on *WSensor*)<br>  – CalculateWWTPEffluentLimits<br>  – CalculateIndustryPermissionLimits |
| Sewage Inspector | • VerifyDischarge<br>  – ObtainLogIndustrialDischarges (Depends on *IndustrialWWRetainer*)<br>  – ObtainLogWWTPInfluents (Depends on *WWReceiver*) |
| WSensor | • ObtainWCharacteristics<br>  – ObtainWPollutantConcentration<br>  – ObtainWVolume |

| | |
|---|---|
| WWTreater | • WWTreatment<br>• DischargeTreatedWW<br>• AchieveAdequatePerformance<br>  – CalculateTreatmentEfficiency<br>    ∗ ObtainInfluentCharacteristics (Depends on *WSensor*)<br>    ∗ ObtainEffluentCharacteristics (Depends on *WSensor*)<br>• LogEffluentCharacteristics (Depends on *WSensor*) |
| WWReceiver | • NegotiateDischarge<br>  – CalculateIndustrialWWDischargePrices<br>    ∗ ObtainTreatementEfficiency (Depends on *WWTreater*)<br>    ∗ ObtainDischargeReferencePrice (Depends on *CompetentAuthority*)<br>  – CalculateIndustrialWWLimits<br>    ∗ CalculateIndustrialWWAvailability<br>      · ObtainMeteoDemandForecast (Depends on *MeteoRetainer*)<br>      · ObtainHouseholdDemandForecast (Depends on *HouseholdWWDischarger*)<br>    ∗ ObtainTreatementEfficiency (Depends on *WWTreater*)<br>    ∗ ObtainWWTPEffluentLimits (Depends on *CompetentAuthority*)<br>• EvaluateInfluentDestination<br>  – ObtainInfluentCharacteristics (Depends on *WSensor*)<br>  – CalculateAvailableCapacity<br>  – ObtainTreatementEfficiency (Depends on *WWTreater*)<br>• LogInfluentCharacteristics (Depends on *WSensor*) |

Table 4.1: Roles and their objectives

The following list describes the different roles in the system. Roles are depicted in **bold** while their objectives are in *italics*. Dependencies with other roles are also presented and. When relevant the dependency relation is also introduced, effectively explaining how the dependant role supports the dependent role.

- **WSensor**: This role is focused on analyzing the characteristics of a water body (*e.g.* a wastewater mass or a water mass in a river) (*ObtainWCharacteristics*). This analysis includes:
  1. Determining concentration of pollutants (Total Suspended Solids (*TSS*), Bio-

chemical Oxygen Demand (*BOD*), Chemical Oxygen Demand (*COD*), Total Nitrogen (*TN*) and Total Phosphorus (*TP*)) (*ObtainPollutantConcentrations*);
2. Measuring volume of water (*ObtainVolume*);

- **IndustrialOperator**: This role is aimed on industrial processes that generate economic revenue and, therefore, allow fulfilling the objective of making profit (*MakeProfit*). These industrial processes produce polluted water masses as a trade-off of their activity (*Produce*). Dealing with this wastewater requires the collaboration of a **IndustrialWWRetainer** which stores the wastewater to be discharged into the sewer system later on (*StoreIndustrialWW*).
- **IndustrialWWRetainer**: Stores wastewater produced by an **IndustrialOperator** (*StoreIndustrialWW*) and takes care of it (*ManageStoredWW*) until it is possible to discharge it to the sewer system (*DischargeIndustrialWW*). To perform such discharge it is required the support of a **IndustrialWWBroker** to negotiate the discharge price and assess the feasibility of discharging some or all the wastewater, or keep storing it. Besides this, it keeps up a registry of all its industrial wastewater discharges performed (*LogIndustrialWWDischargeCharacteristics*), which requires the support of a **WSensor** to analyze the characteristics of discharged wastewater. This information can be used by a **SewageInspector** to verify $\mathcal{I}$ are properly managing discharges (*VerifyDischarge*).
- **IndustrialWWBroker**: Negotiates industrial wastewater discharges with a **WWReceiver** to assess how much wastewater is feasible to be discharged (*AssessAmountOfIndustrialWWDischarge*). From the **IndustrialWWBroker** perspective this assessment requires knowing its reserved cost (*ObtainDischargeReservedCost*) (*i.e.*, how much $I_i$ is willing to pay according to the discharge price given by the **WWReceiver**). The **IndustrialWWBroker** will compute this reserved cost depending on:
  - The above mentioned discharge price requested to the **WWReceiver** (*ObtainDischargeReceiverPrice*);
  - The reference prices given by the **CompetentAuthority** with regards to volume and pollutant concentration discharges (*ObtainDischargeReferencePrice*);
  - The characteristics (*i.e.*, volume and pollutant concentration) of the wastewater to be discharged, which are provided by the **IndustrialWWRetainer** as part of the wastewater discharge negotiation process.

  Once negotiation ends, it can happen either:
  - An agreement is not achieved and wastewater is kept stored until prices go down (if storage capacity allows to do so);
  - An agreement is achieved and some (or all) wastewater is discharged; it is possible that the actual discharge complies or not to what was previously agreed.

  **IndustrialWWBroker** is consuming treatment capacity to discharge wastewater and comply with the policies and norms that regulate wastewater discharges and ensure water quality. **WWReceiver** is offering such service thus the relation is a consumer-provider one thus the dependency between both roles in the *role dependency* diagram is a market dependency.
- **MeteoRetainer**: Stores water coming from meteorological conditions (*e.g.*, rain) (*StoreMeteoWW*). This stored water is analysed (*ObtainMeteoWWCharacteristics*) and its destination is assessed (*AssessMeteoWWDestination*) (*e.g.*, the storage tank is overflown and it cannot retain more water). Depending on the assessment result water is either:

  - Discharged to a **WWReceiver** (*DischargeMeteoWW*);
  - Bypassed to the river (*BypassMeteoWW*), which also requires informing the **CompetentAuthority** of such event (*InformBypassMeteoWW*);
  - Keep it stored, if possible.

  The relation between *MeteoRetainer* and *WWReceiver* as seen in Figure 4.2 is a network dependency. Both roles act as equals since none of them have an authoritative power over the other nor they are consuming any service provided in a market. This relation is motivated by the willingness to coordinate their actions and achieve the system's goal of providing a certain level of water quality which emanates from the competent authority goals.
- **HouseholdWWDischarger**: This role generates wastewater produced by residential areas and discharges it to the sewage system (*DischargeHouseholdWW*). It provides the wastewater characteristics that it generates to roles that may need it (*ObtainHouseholdWWCharacteristics*) such as **WWReceiver** as part of its goal *ObtainHouseholdDemandForecast*.
- **WWReceiver**: Takes care of negotiating the reception of wastewater masses (*NegotiateDischarge*). This includes providing discharge prices for $\mathcal{I}$ (*CalculateIndustrialWWDischargePrices*) and the treatment capacity available for industrial wastewater (*CalculateIndustrialWWAvailability*). To calculate discharge prices the **WWReceiver** uses discharge reference prices provided by the **CompetentAuthority**, the current treatment efficiency in $wwtp_j$ (*ObtainTreatmentEfficiency*) as well as the characteristics of the industrial wastewater that $I_i$ wants to discharge, which are provided by the **IndustrialWWRetainer** as part of the wastewater discharge negotiation process. Concerning treatment capacity availability, it depends on:
  - Wastewater being received from households (*ObtainHouseholdDemandForecast*);
  - Meteorological retainers status (*ObtainMeteoDemandForecast*);
  - Current efficiency in the **WWTreater** (*ObtainTreatmentEfficiency*;
  - Current $wwtp_j$ effluent limits imposed by the **CompetentAuthority** (*ObtainWWTPEffluentLimits*).

  Once influent is received, the **WWReceiver** determines its destination, either to be sent for treatment or bypass it directly to the river. This decision depends on (*EvaluateInfluentDestination*) wastewater characteristics (*ObtainInfluentCharacteristics*), $wwtp_j$ current treatment efficiency (*ObtainTreatmentEfficiency*) and available capacity (*CalculateAvailableCapacity*). Finally, it also keeps a record of the influent characteristics received (*LogInfluentCharacteristics*) for the **SewageInspector** (as part of the *VerifyDischarge* task) and **WWTreater** (in order to calculate treatment efficiency).
- **WWTreater**: Processes the wastewater to reduce its pollutants concentration (*WWTreatment*). Once the treatments ends, treated water is discharged as an effluent to the river. This effluent is analyzed (*ObtainEffluentCharacteristics*) and information is logged so the **CompetentAuthority** can audit it (*VerifyWWTPEffluent*). Given the effluent and influent characteristics (*ObtainInfluentCharacteristics / ObtainEffluentCharacteristics*), **WWTreater** can calculate treatment efficiency (*CalculateTreatmentEfficiency*). This calculation is used to keep $\mathcal{W}$ as efficient as possible (*AchieveAdequatePerformance*). It is also used to support **WWReciever** during discharge price negotiation described before.
- **CompetentAuthority**: Takes care of ensuring the river has a certain quality level determined by applicable legislation (*EnforceWaterQualityPolicies*). To do so, it assess

river water quality continuously (*AssessRiverWaterQuality*) by means of collecting river sensor data (*CollectRiverSensorData*); this data is requested to entities playing the **WSensor** role. Such entities are physically located along the river basin. According to the results of this assessment it may request a **SewageInspector** to verify $I_i$ are compliant to what they agreed with $wwtp_j$ concerning wastewater discharges (*VerifyDischarge*).

Depending on the current status or even exceptional meteorological conditions, it is possible that the **CompetentAuthority** requires to establish exceptional restrictions on the system (*EstablishExceptionalRestrictions*) to ensure river water quality. The **CompetentAuthority** also uses the above mentioned assessment for two main purposes: First, providing discharge limits for $\mathcal{I}$ (*CalculateIndustryPermissionLimits*) and effluent limits for $\mathcal{W}$ (*CalculateWWTPEffluentLimits*). In this way **CompetentAuthority** can establish pollutant reductions in certain river sections and, in general, in the river as a whole. Second, determining reference prices for wastewater discharges (*CalculateDischargeReferencePrices*).

These elements are used by both **IndustrialWWBroker** and **WWReceiver** as part of their negotiation process when discharging industrial wastewater.

Finally, **CompetentAuthority** also checks $\mathcal{W}$ are complying with effluent limits established (*VerifyWWTPEffluent*). Given that the competent authority has to ensure that the rest of roles are compliant with water quality legislation, all the dependencies that are related to this role are hierarchical ones, since the system requires the competent authority to have a power relation on all the roles to oblige them coordinate appropriately. Given the ascendancy role of the competent authority over the other roles.

- **SewageInspector**: Monitors activity in the urban sewage system, verifying industrial discharges (*VerifyDischarge*) that are being produced and checking that the influents received by the **WWReceiver** (*ObtainLogWWTPInfluents*) and effluents sent by **IndustrialWWRetainer** (*ObtainLogIndustrialDischarges*) match and there is no such a strong deviation that place $\mathcal{W}$ into a hazardous or unsuitable state (*e.g.*, too many industries discharging more water or with more pollutant concentration that the ones agreed, thus affecting $\mathcal{W}$ processes negatively). The result of this verification is reported to the **CompetentAuthority** to act accordingly.

### 4.3.2 Interaction Structure

In *ALIVE* the *Interaction Structure* allows the description of abstract patterns of interaction which are the way the roles coordinate their behavior, managing their dependencies while they pursue their individual and collective objectives. The interaction structure (see Figure 4.3) defines interaction patterns known as *scenes* [DDPVS09] that allow actors to coordinate. The structure defines a set of scenes and transitions among them. On every scene one or more role dependencies (identified in the previous phase) are managed. Table 4.2 summarizes the role dependencies shown at every particular scene.

The structure's entry point is represented by a circle (*init* label), while the exit points are represented by triangles (*end* label). Scenes are represented by rectangles and connected by lines (scene transition arcs) that allow the system to navigate from scene to scene. Inside every scene, the landmark patterns describe the protocol that must be used to achieve the scene result[DDPVS09]. This diagram will focus on *scene transition*, allowing the following diagrams to focus on the different particular scenes by showing the landmark patterns inside them. The entry point leads to the different wastewater generation scenes (*Household*, *MeteoWWGenerate* and *IndustryWWGenerate*) and the scenes related to water quality protection by the competent authority (*EnforceWaterQualityPolicies*, *ComputeWWTPEffluentLimits*, *ComputeIndustryPermissionLimits* and *ComputeDischargeReferencePrices*). *HouseholdWWGenerate* leads directly to wastewater treatment scenes. *MeteoWWGenerateScene* contain particular scenes for discharging the water to $\mathcal{W}$ (*MeteoWWDischarge*) or bypassing it (*MeteoWWBypass*). Industrial production is divided in two parts: first wastewater is generated and stored *IndustryWWGenerate* and, later on, a negotiation to discharge wastewater in tank is done (*IndustryManageWWTank*) where a price to discharge is formed (*WWTPAssessDischargePrice*); depending on the negotiation result (*IndustryAssessWWDischarge*) wastewater is discharged (*IndustryWWDischarge*) or it is kept.

Some discharges will have a discharge verification performed asynchronously by the competent authority (*VerifyDischarge*). Wastewater treatment scenes include receiving influent (*WWTPReceiveInfluent*) and either treating (*WWTPTreatInfluent*) or bypassing it (*WWTPBypassfluent*)).

| **Dependency** | **Scenes** |
|---|---|
| CollectRiverSensorData | - EnforceWaterQualityPolicies<br>- ComputeWWTPEffluentLimits |
| LogIndustrial WWDischargeCharacteristics | - IndustryWWDischarge |
| LogInfluentCharacteristics | - WWTPReceiveInfluent |
| ObtainEffluentCharacteristics | - WWTPTreatInfluent |
| ObtainHouseholdWWCharacteristics | - Household |
| ObtainMeteoWWCharacteristics | - MeteoWWGenerate |
| ManageIndustrialWW | - IndustryWWGenerate |
| ManageStoredWW | - IndustryManageWWTank<br>- IndustryAssessWWDischarge |
| ObtainDischargeReferencePrice | - IndustryAssessWWDischarge |
| ObtainDischargeReceiverPrice | - IndustryAssessWWDischarge |
| ObtainWWTPEffluentLimits | - WWTPAssessDischargePrice |
| VerifyDischarge | - VerifyDischarge |

| | |
|---|---|
| ObtainTreatementEfficiency | - WWTPAssessDischargePrice |
| ObtainInfluentCharacteristics | - WWTPTreatInfluent |
| LogEffluentCharacteristics | - WWTPTreatInfluent<br>- WWTPBypassfluent |
| ObtainEffluentCharacteristics | - WWTPTreatInfluent |
| ObtainIndustrialWWCharacteristics | - WWTPAssessDischargePrice |
| AssessMeteoWWDestination | - MeteoWWGenerateScene |
| ObtainIndustrialWWCharacteristics | - IndustryManageWWTank<br>- WWTPAssessDischargePrice |
| ObtainMeteoDemandForecast | - WWTPAssessDischargePrice |
| ObtainHouseholdDemandForecast | - WWTPAssessDischargePrice |
| ObtainLogWWTPInfluents | - VerifyDischarge |
| ObtainLogIndustrialDischarges | - VerifyDischarge |
| InformBypassMeteoWW | - MeteoWWBypass |

Table 4.2: Scenes and role dependencies in them

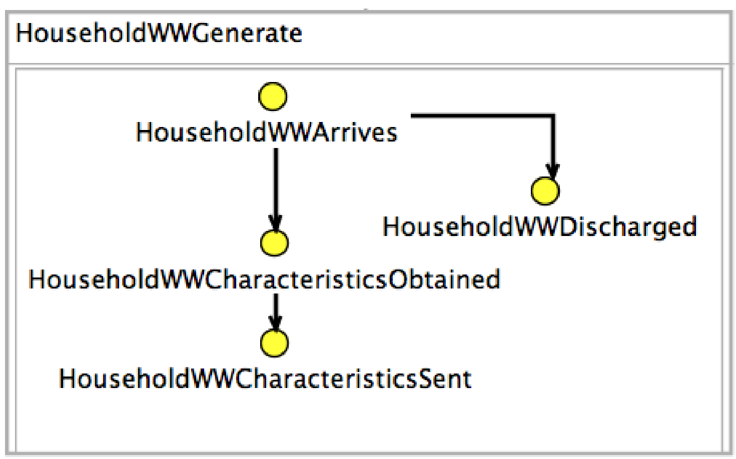



Figure 4.4: Landmark patterns for the household discharge

Figure 4.4 depicts the internal landmark patterns for the *HouseholdWWGenerate*. It shows the flow followed by a household wastewater discharge. Household wastewater lacks retention tanks, and therefore is treated as a continuous wastewater flow. As household wastewater is generated (*HouseholdWWArrives*) it is discharged into the treatment plant (*HouseholdWWDischarged*). In parallel, water characteristics are analyzed (*HouseholdWWCharacteristicsObtained*) and provided to the competent authority (*HouseholdWWCharacteristicsSent*).

Figure 4.5 depicts the internal landmark patterns for the *MeteoWWGenerate*, *MeteoWWDischarge* and *MeteoWWBypass* scenes. It shows how meteorological wastewater is


Figure 4.3: Diagram of the roles and role dependencies

generated and temporary stored to be either discharged for normal processing or bypassed to avoid an overflow. Meteorological water is potentially polluted and usually generated in concentrated intervals (specially in the Mediterranean area), therefore retention tanks are available to store it. Wastewater is generated ($MeteoWWArrives$), stored ($MeteoWWStored$) and analysed ($MeteoWWCharacteristicsObtained$). Depending on arriving wastewater characteristics and tank level three options are available, landmark *MeteoWWDestinationAssessed* represents the process of selecting one of these options:

- Tank is discharged to the $wwtp_j$ (either partially or totally). Represented by scene *MeteoWWDischarge* where tank is discharged (*MeteoWWDischarged*) and tank characteristics are communicated to the competent authority (*MeteoWWCharacteristicsSent*)

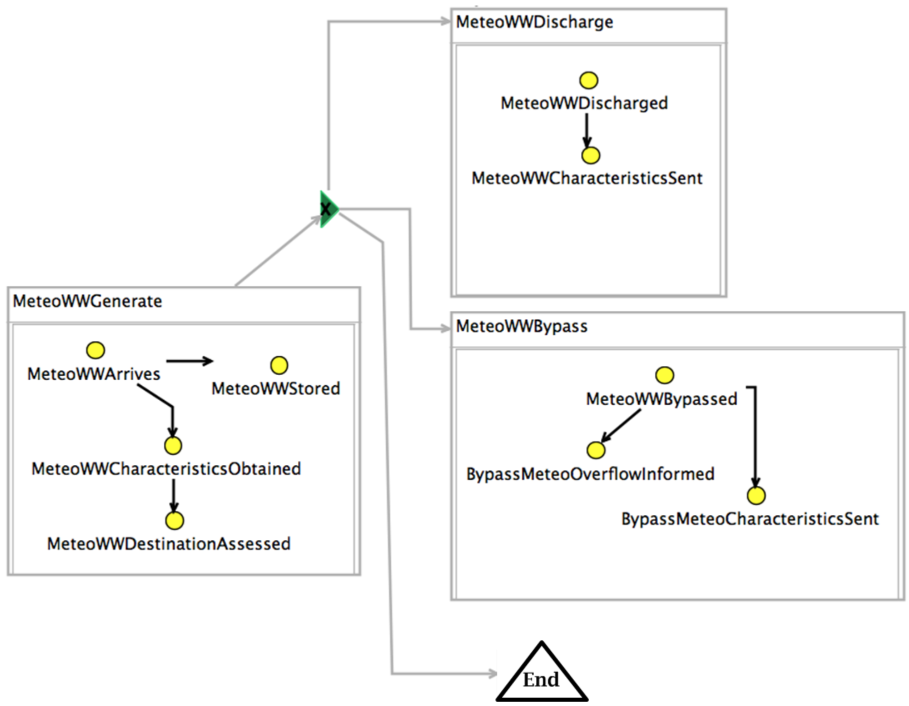



Figure 4.5: Landmark patterns for the meteorology discharge

.

- Tank is bypassed to the river. Represented by scene *MeteoWWBypass* where tank is discharged in the presence or to prevent an incoming overflow ($MeteoWWBypassed$) and the competent authority is informed of this exceptional situation ($BypassMeteoOverflowInformed$) and the bypass characteristics (*Bypass-MeteoCharacteristicsSent*). This scene is reached when concentrated rain bypasses the capacity of the conveyor unit connecting the tank with $wwtp_j$ and a bigger conveyor unit with higher capacity (connecting directly to the river) must be used.
- Do nothing. Tank is not full yet and $\mathcal{W}$ are busy treating water from $I_i$ or households. Water is retained in the tank and a final state is reached.

Figure 4.6 depicts the internal landmark patterns for the *IndustryWWGenerate* and *IndustryManageWWTank* scenes. It depicts how industrial wastewater is produced and temporary stored while Industry agent negotiates with WWTP when to perform a discharge. Industry water is stored in retention tanks, so $\mathcal{I}$ can perform a negotiation process with $\mathcal{W}$ to see which is the best moment for treating wastewater (*e.g.*, looking for valley hours where the plant is idle). As a result of industry's production process (*Produce*) industrial wastewater is generated (*IndustrialWWGenerated*) and stored in retention tanks. As tanks have a limited capacity, they have to be managed (*IndustrialWWManaged*) finding out the best moment for releasing their contents into the WWPTs. In order to find out the cost of treating generated wastewater, when wastewater is available on the tank (*IndustrialWWIn-*

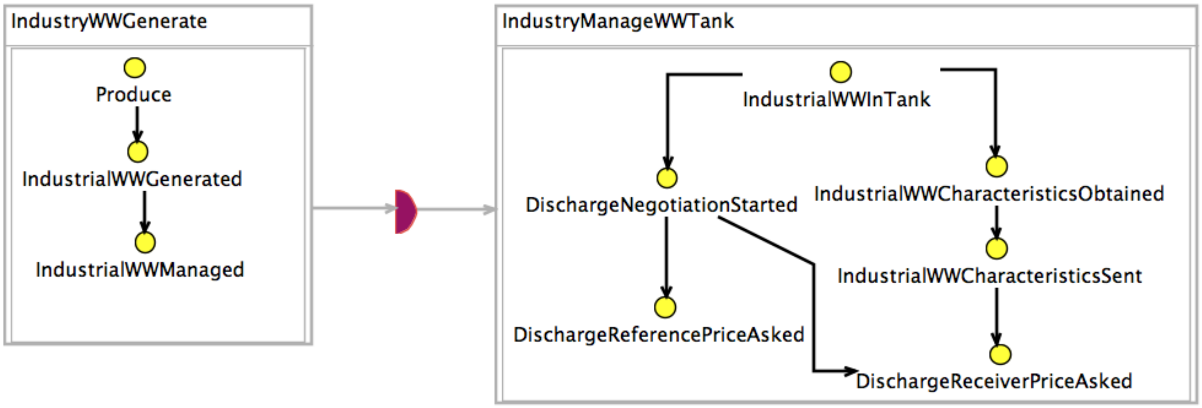



Figure 4.6: Landmark patterns for the industry waste generation

*Tank*), the water is analyzed (*IndustrialWWCharacteristicsObtained*). In parallel, $\mathcal{I}$ start a negotiation with $\mathcal{W}$ to find out the discharge receiver price (*DischargeNegotiationStarted*) and obtain the reference price from the competent authority (*DischargeReferencePriceAsked*). The negotiation requires wastewater characteristics to be provided to $wwtp_j$ (*Industrial-WWCharacteristicsSent* so the discharge receiver price can be effectively asked (*Discharg-eReceiverPriceAsked*).

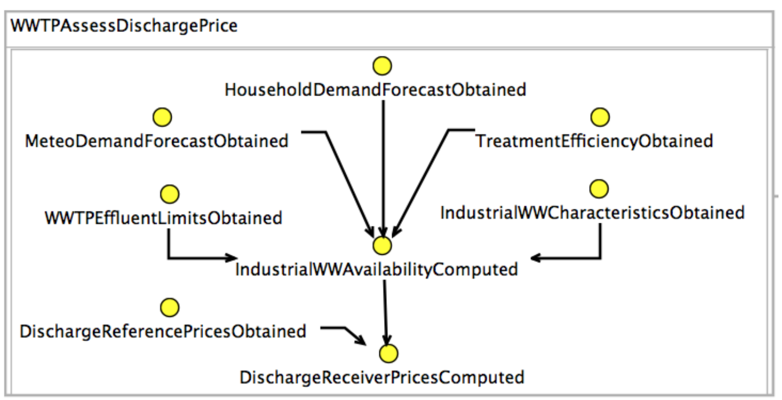



Figure 4.7: Landmark patterns for the industry price negotiation

Figure 4.7 depicts the internal landmark patterns for the *WWTPAssessDischargePrice* scene. It shows how the price to handle a industrial discharge is determined according to household and meteorological demand forecasts, limitations imposed by norms, volume availability and the wastewater characteristics. The price is determined by finding out the actual $wwtp_j$ treatment capacity available for $\mathcal{I}$ and putting it in contrast with the water characteristics provided by the $I_i$ asking the price. Knowing effluent quality required by the competent authority (*WWTPEffluentLimitsObtained*), the actual plant's efficiency (*TreatmentEfficiencyObtained*) and the characteristics of the wastewater to be treated (*IndustrialWWCharacteristicsObtained*) allows the plant to plan ahead and obtain a water treatment estimation cost. However, typically not all the treatment capacity is available for $\mathcal{I}$, as $\mathcal{W}$ have also to deal with household (*HouseholdDemandForecastObtained*) and meteorological wastewaters (*MeteoDemandForecastObtained*). Therefore, the plant asks for forecasts on these two wastewater sources to take them into account in the planning, effectively providing a more accurate estimated cost for cleaning the industrial wastewater by taking into account plant's availability for treating it (*IndustrialWWAvailabilityComputed*). Putting in contrast the cost with the discharge reference prices provided by the competent authority (*DischargeReferencePricesObtained*) allows the plant to send a final price to the $I_i$ (*DischargeReceiverPricesComputed*).

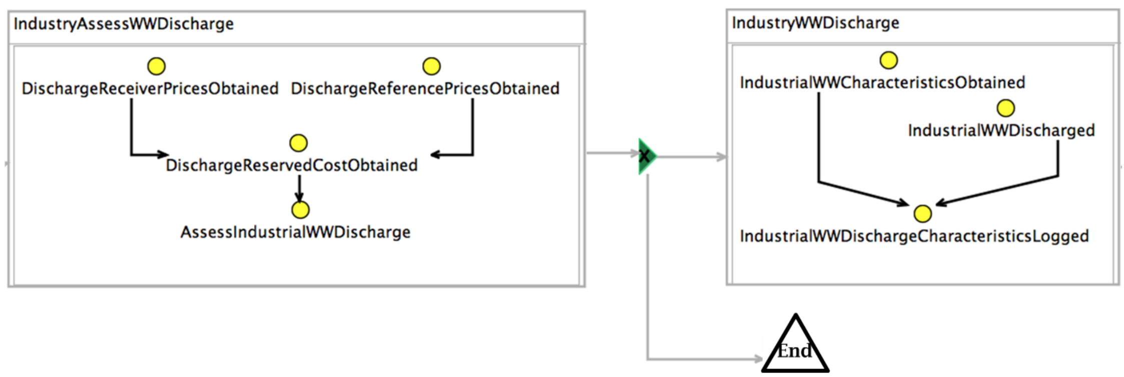



Figure 4.8: Landmark patterns for the industry discharge

Figure 4.8 depicts the internal landmark patterns for the *IndustryAssessWWDischarge* and *IndustryWWDischarge* scenes. It shows the decision process carried out by Industry agent to manage its wastewater: keeping it stored or discharge it so a WWTP can process it. Such decision depends on the processing price and the agent's reserved cost. The first scene combines discharge reference and receiver prices (*ObtainDischargeReceiverPrices* and *ObtainDischargeReferencePrices*) to obtain a discharge reserved cost (*DischargeReservedCostObtained*). The cost, allows to decide weather it is worth it or not to clean wastewater stored in $I_i$ retention tank (*AssessIndustrialWWDischarge*). In case it is not worth it, a final state is reached. If it is worth treating the wastewater, a new scene is visited *IndustryWWDischarge* where wastewater characteristics are obtained (*IndustrialWWCharacteristicsObtained*), water discharged to the treatment plant (*IndustrialWWDischarged*) and water characteristics logged (*IndustrialWWDischargeCharacteristicsLogged*).

Figure 4.9 depicts the internal landmark patterns for the *WWTPReceiveInfluent* and *VerifyDischarge* scenes. Upper diagram shows the process of receiving a wastewater mass into a WWTP and deciding if it can be effectively processed or it has to be bypassed. Lower diagram shows how logged information (influents previously received) is double-checked against agreed industrial discharges. The first scene starts by receiving wastewater ($WWArrives$) from an industry, household or meteorological tank and analyzing it

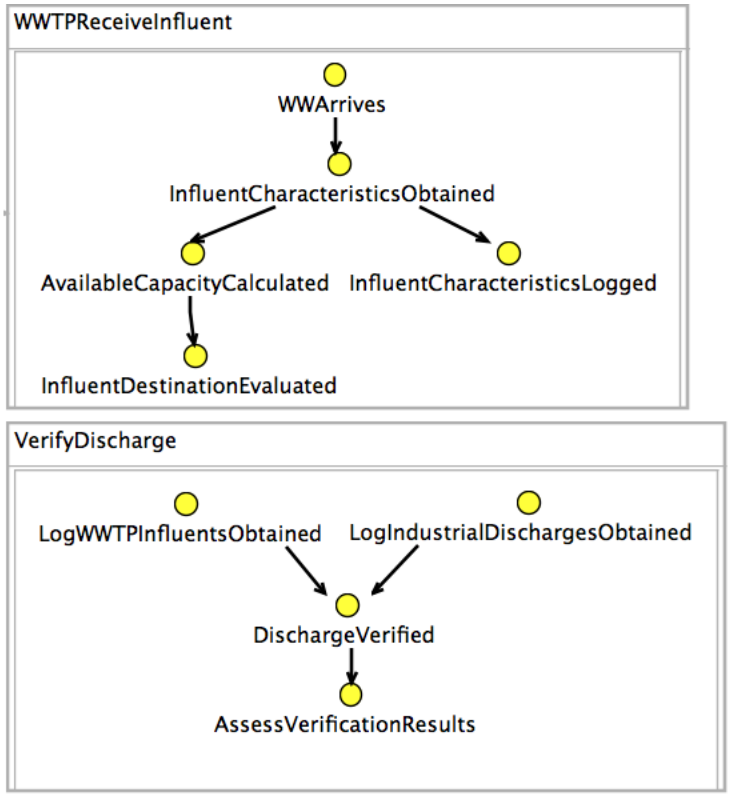



Figure 4.9: Landmark patterns for WWTP receive influent

($InfluentCharacteristicsObtained$). Knowing influent characteristics and $wwtp_j$ available capacity (*AvailableCapacityCalculated*) a decision (*InfluentDestinationEvaluated*) on weather to treat the influent or bypass it can be taken. In parallel, influent characteristics are logged (*InfluentCharacteristicsLogged*). This step is essential for detecting when a particular industry $\mathcal{I}$ has reached an agreement to provide an amount of water with a pollutant concentration to be treated for a particular price and has failed to fulfill the agreement (*e.g.*, sending more wastewater or with higher pollution concentration). In order to detect and sanction this undesirable behavior, the *VerifyDischarge* scene is available. It consists in the Sewage Inspector asynchronously checking that both volume and composition (*e.g.*, pollutant concentration) declared by industries (*LogIndustrialDischargesObtained*) match the actual volume and composition received by $wwtp_j$ (*LogWWTPInfluentsObtained*).

Results of the inspection (*DischargeVerified*) are notified to the competent authority. Competent authority analyses results *AssessVerificationResults* so corrective actions can be taken (*e.g.*, sanctioning a particular industry $\mathcal{I}$ with a fine) and incentives applied (*e.g.*, providing lower reference prices to industries that always comply with discharge inspections).

Figure 4.10 depicts the internal landmark patterns for the *WWTPTreatInfluent*. It shows

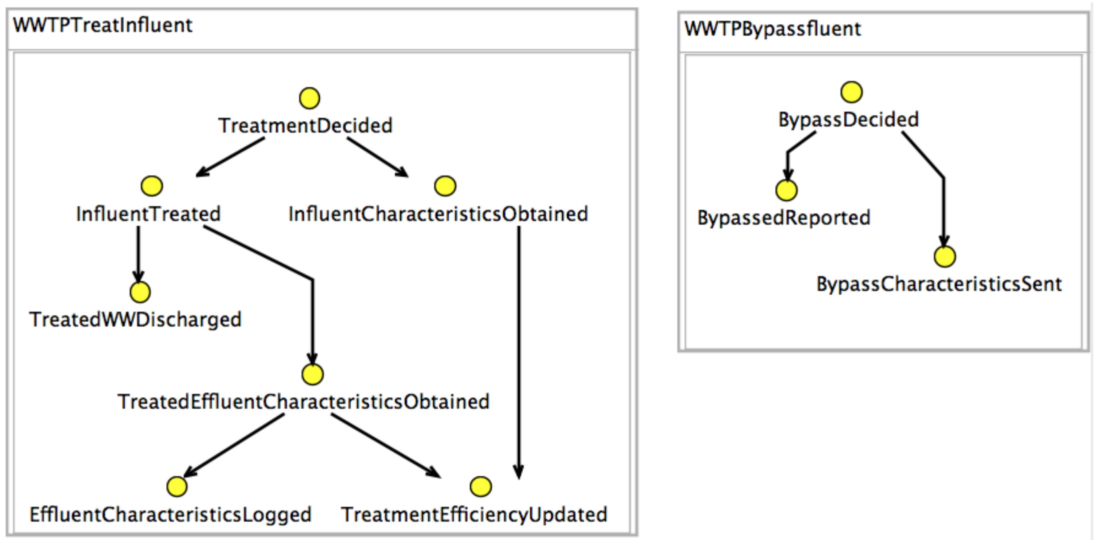



Figure 4.10: Landmark patterns for WWTP influent treatment

how wastewater is processed in a WWTP. The left diagram shows the treatment process carried out by WWTP agent: the influent is treated and the effluent characteristics and process efficiency is logged. The right diagram shows how wastewater is bypassed and directly thrown to the river; such bypass has to be reported to the Competent Authority. On the one hand, if influent is treated the scene *WWTPTreatInfluent* is visited, where the following landmarks are achieved: influent treatment has been decided (*TreatmentDecided*) so influent is treated (*InfluentTreated*) and discharged (*TreatedWWDischarged*). In parallel, influent is analyzed (*InfluentCharacteristicsObtained*). Once the treated water has been discharged, effluent is also analyzed (*TreatedEffluentCharacteristicsObtained*) so influent and effluent characteristics can be contrasted updating treatment efficiency statistics (*TreatmentEfficiencyUpdated*). In parallel, effluent characteristics are logged (*EffluentCharacteristicsLogged*) making them available to the competent authority to verify $wwtp_j$ effluents are compliant to the limits established.

On the other hand, if influent is bypassed (*BypassDecided*), typically when it exceeds $wwtp_j$ capacity, the competent authority is informed (*BypassedReported*) and bypass characteristics provided (*BypassCharacteristicsSent*). Competent authority is warned so actions to protect water quality can be taken if they are required down the river (*e.g.,* modify industry $\mathcal{I}$ permissions or $wwtp_j$ effluent limits).

Figure 4.11 depicts the internal landmark patterns for the *EnforceWaterQualityPolicies*, *ComputeWWTPEffluentLimits*, *ComputeIndustryPermissionLimits* and *ComputeDischargeReferencePrices* scenes. It shows the different processes carried out by the Competent Authority: upper-left describes how water quality is assessed and restrictions are enabled to ensure water quality policies. Upper-right show how the effluent limitations for WWTP are determined. Lower-left diagram depicts how industrial permits are calculated according to water policies and current river status. Bottom-right diagram shows how discharge


Figure 4.11: Landmark patterns for competent authority control policies

references prices are determined to be used later on in the negotiation process between industries and WWTPs. The scene *EnforceWaterQualityPolicies* starts by collecting river data (*RiverSensorDataCollected*) as input from multiple sources (river sensors, $\mathcal{W}$ effluents and influents, industry effluents, actor notifications, *etc.*) which is provided to the competent authority (*RiverSensorDataSent*). Data is put in contrast with water quality policies (*WaterQualityPoliciesObtained*) effectively assessing river quality (*RiverQualityAssesed*). Finally, the scene is detecting exceptional situations and applying the appropriate restrictions *ExceptionalRestrictionsUpdated*. For instance, if a meteorological overflow is notified, industries can not discharge wastewater to the river and $\mathcal{W}$ must close until the overflow is solved. In order to tackle these restrictions we will use *ALIVE*'s normative structure, grounded on deontic norms (that specify actor's obligations and permissions). Due to the complexity of this normative model, we will present it in a separate section (see §4.5). The scenes *ComputeWWTPEffluentLimits* and *ComputeIndustryPermissionLimits* put river data in contrast with environmental and river quality policies (*RiverSensorDataCollected*, *RiverSensorDataSent* and *WaterQualityPoliciesObtained*) and obtain new $\mathcal{W}$ effluent and $\mathcal{I}$ permission limits (*WWTPEffluentLimitsCalculated* and *IndustrialPermissionLimitsCalculated*).

Finally, both $\mathcal{W}$ effluent limits (*WWTPEffluentLimitsUpdated*) and $\mathcal{I}$ permission limits (*IndustrialPermissionLimitsUpdated*) are updated. These scenes allow for analyzing river water quality at real-time and putting in place measures to correct deviations in desired river quality. Scene *ComputeDischargeReferencePrices* puts is contrast river sensor data (*RiverSensorDataCollected*) with water quality policies (*WaterQualityPoliciesObtained*) to precompute discharge reference prices (*DischargeReferencePricesComputed*). On the one hand, this allows to adapt discharge reference prices to the state of the river (interpreted from river sensor data) and the water quality policies. On other hand, this allows to provide a swift answer when discharge reference prices are asked by $\mathcal{I}$ or $\mathcal{W}$.

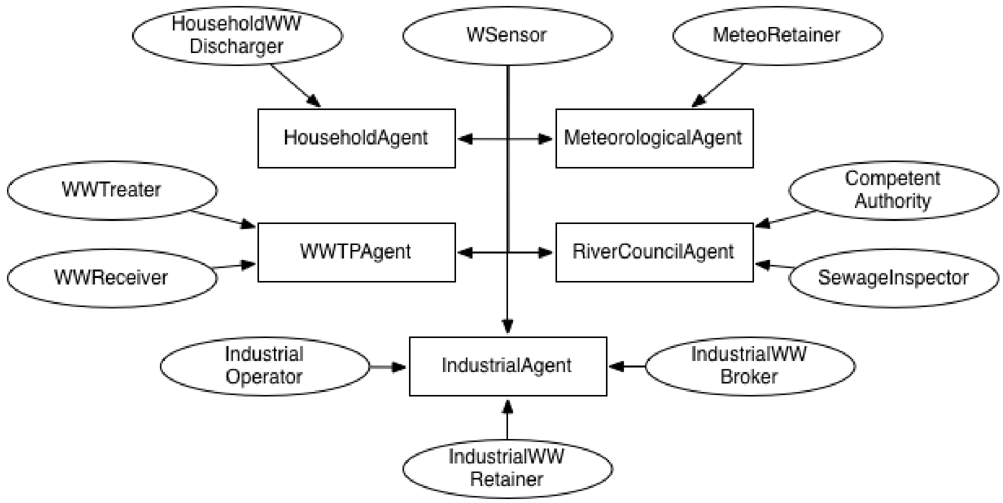



Figure 4.12: Roles that each kind of agent can enact

### 4.3.3 Social Model

Up to this point the overall system has been defined in terms of roles, their aims and dependencies, without taking into account who will actually enact those roles at run-time. The *Social Model* defines the way concrete agents enact the organization roles, guiding their behaviour and achieving coordinated action by following the organizational patterns established for the roles they enact. An agent can enact one or several roles, depending on their objectives, and one role can be enacted by more than one agent. In the case study, the objectives of UWS infrastructures act as the main driving force for allocating roles into agents.

Figure 4.12 shows the model of agents. It describes the roles that each kind of agent can enact during its lifetime. The central axis is composed by the types of agents with a *Household*, *Meteorological*, *Industrial*, *WWTP* and *River Council agents* (square-shaped). The surrounding nodes represent the roles (round-shaped). The arrows connect each role with the agents type that performs it.

From the model of roles, agents, and the scenario defined we can enumerate the following relevant elements:

- the set of WWTPs (Receivers and treaters) as $[W_1, \dots W_k] \in \mathcal{W}$
- the set of Industries (Operators, retainers and brokers) as $[I_1, \dots I_k] \in \mathcal{I}$
- the set of Households as $[\kappa_1, \dots \kappa_k] \in \mathrm{K}$
- the set of Meteorological retainers as $[\psi_1, \dots \psi_k] \in \Psi$
- the set of Water Treatments as $[T_1, \dots T_k] \in \mathcal{T}$
- the set of Water Masses as $[M_1, \dots M_k] \in \mathcal{M}$
- the set of Water Samples as $[S_1, \dots S_k] \in \mathcal{S}$
- the set of Water Pollutants as $[O_1, \dots O_k] \in \mathcal{O}$
- the set of Water Sensors as $[R_1, \dots R_k] \in \mathcal{R}$

- the set of competent authority representatives (*i.e.* sewage inspectors) as $[N_1, \dots N_k] \in \mathcal{N}$
- a competent authority as $C$

## 4.4 Agents: Behaviour and decision-making

The most important roles in our system, from an integrated management perspective, are two. First, the *Competent Authority* is responsible for all matters relating to the collection, treatment and disposal of wastewater. Second the $wwtp_k$ which is responsible for treating wastewaters in that sector or city. In this section we focus on the latter to describe the decisions it has to make to carry out the negotiation with industries that would like to discharge the wastewater resulting from their activities. The discharge of a wastewater mass requires an *agreement* between a particular $I_i$ and a particular $wwtp_j$ that can accept it for treatment. This process requires *knowing* if the $wwtp_j$ is capable to properly handle the proposed wastewater discharge.

We define the water characteristics of a water mass $W$ as the pair $(V, C)$, where: $V$ is the volume of water mass given in cubic meters $m^3$ and, $C$ is the pollutant concentration in the water mass. We also define $C = \left(C^1, \dots, C^r\right)$ as the set of pollutants concentrations[1] where each $C^i$ corresponds to a specific pollutant concentration $(1 \leq i \leq r)$. In our system, as already explained in §4.2.1, we consider five different pollutants: $(TSS, BOD, COD, TN, TP)$. Thus $C = \left(C^1, C^2, C^3, C^4, C^5\right) = (TSS, BOD, COD, TN, TP)$ [VCP12].

Given this premises, the *negotiation* process between $I_i$ and a $wwtp_j$ can be described as follows: The $wwtp_j$ checks if it can *manage* the wastewater that the industry wants to discharge (a wastewater mass characterized as $(V_i, C_i)$). This means ensuring there is enough physical space to receive it (*volume availability*) and that the plant can effectively treat the pollutants contained in the wastewater mass (*pollutant concentration admissibility*); if wastewater contains a high pollutant concentration it can harm the treatment process since it depends on bacteria colonies that may perish.

Volume availability ($V_{available}$) depends on the design volume of the $wwtp_j$ ($V_{capacity}$), the amount of domestic wastewater sent by households ($V_d$) and meteorological phenomena ($V_m$)(*e.g.*, rain), whose treatment is mandatory. Finally, previously agreed industrial discharges ($V_{scheduled}$) have to be taken into account to know what volume capacity remains available for new industrial discharges:

$$V_{available} = V_{capacity} - V_d - V_m - V_{scheduled} \tag{4.1}$$

Therefore, if $V_i \leq V_{available}$ then there is enough space in the $wwtp_j$ to accept the wastewater mass.

To verify pollutant concentration admissibility the process is similar although $wwtp_j$ can admit a higher concentration than the one it can effectively manage; however, this will imply a significant higher cost for the industry. Pollutant concentration admissibility depends on how much pollutant concentration $wwtp_j$ can manage as a parameter design of the plant ($C^r_{admissible}$). Thus, if $C^r_{admissible} \geq C^r_i$ the wastewater mass will be accepted without extra cost. Otherwise the pollutant overload will carry an extra cost to the price

[1] Pollutant concentration is given in $(\frac{kg}{m^3})$.

that industry has to pay. This price is calculated by $wwtp_j$ as follows:

$$P(WW_i, wwtp_j) = P((V_i, C_i), wwtp_j) = VC(V_i, wwtp_j) + PC(WW_i) \tag{4.2}$$

In equation 4.2 is divided into two parts: a volumetric cost and a pollutant cost. The first cost represents the cost of accepting a certain volume of wastewater generated by $I_i$ according to the current state of the $wwtp_j$ as well as the taxes defined by the competent authority. The second cost computes the cost of processing the wastewater discharged by industry $I_i$ according to its pollutant concentration.

Volumetric cost is defined as:

$$VC\left(V_i, wwtp_j\right) = UC(V_i) \cdot \left(1 + \frac{V_{ocWWTP}}{V_{totWWTP}}\right)^x \tag{4.3}$$

The first part of equation 4.3 is developed as:

$$UC\left(V_i\right) = V_i \cdot \left[GT + ST\left(i\right)\right]; \tag{4.4}$$

$GT$ corresponds to the General Tax paid by industries while $ST(i)$ is the Specific Tax that depends on the type of industry where $I_i$ belongs to. This formula expresses the taxes paid by an industry to discharge a certain wastewater volume ($V_i$). These taxes are defined by the *Competent Authority*. The second part of 4.3 is an exponential function that measures how much volume is already occupied in the $wwtp_j$ from its total capacity. The potential factor, *x*, expresses the fact that the price increases as the $wwtp_j$ gets fuller.

Pollutant cost is defined as:

$$PC(WW_i) = PC((V_i, C_i)) = V_i \cdot \left(\sum_{r=1}^{5} C_i^r \cdot q^r \cdot g_i^r\right) \tag{4.5}$$

Where:

- $q^r$ is the reference price for discharging a kilogram of pollutant $r$ and is determined by the competent authority;
- $g_i^r$ is a peak coefficient that expresses the deviation between the pollutant $r$ concentration in the wastewater mass sent by industry $I_i$ and the pollutant concentration limits agreed with the competent authority.

The coefficient $g_i^r$ basically allows industries to deal with unexpected situations that oblige them to discharge more pollutant concentration that they are allowed to, by paying a higher price for doing so. It is calculated as $g_i^r = \frac{C_i^r}{X_i^r}$, where $X_i^r$ corresponds to the maximum concentration of pollutant $r$ in a discharge, agreed between industry $I_i$ and the competent authority.

In this way, equation 4.2 is taking into consideration the permits given by the competent authority to industries in terms of allowed pollutant concentration as well as taxes for discharging wastewater, and the current status of the $wwtp_j$ (in terms of volume occupancy). It also allows industries to overpass pollutant concentration to deal with unexpected situations as long as they are capable to pay for the extra cost it implies.

Finally, the $wwtp_j$ communicates the price and available volume to the industry. Then $I_i$ decides either to keep storing the wastewater in its internal storage tanks or proceed with a full or partial discharge.

### 4.4.1 Negotiation

We use Autonomous Agents to cope with the problem in view of the various, sometimes conflicting, goals that the identified actors have to fulfil their private interests. Within this frame, agents are able to *negotiate* between them. Negotiation is a means for agents to communicate and compromise to reach mutually beneficial agreements [FWJ04]. Specifically, in this *Chapter*, we address the negotiation of $wwtp_j$ with several industries that would like to discharge the wastewater resulting from their activities. As it has been described above there are, at least, two conflicting variables described by equations 4.1 and 4.2. There are also issues related with time, like the time of service delivery (*e.g.* when the discharge can be done). Also, there is a set of norms and rules to be followed by all agents. It is possible to address the problem as a multi-issue negotiation process. In a real scenario, issues are constrained each other and this makes agents' utilities nonlinear. Furthermore, even in collaborative situations, in order to reach an agreement, agents need to act competitively because of their self-interested nature [IHK07].

However, agent negotiation is a complex scenario, offering a wide range of options to consider and decisions to make at every negotiation step. In order to tame this complexity, negotiation protocols can be used. Such protocols specify which are the valid options at every negotiation step. By providing a structured interaction with predictable outcomes and limiting the options to be taken at every point of time, negotiation protocols will effectively simplify agent reasoning. There is a wide range of negotiation protocols available [PCJ04].

For this environment we are proposing a negotiation protocol based on the FIPA Iterated Contract Net Interaction Protocol [FIP00], which is an extension of the Contract Net Interaction Protocol. The protocol starts when an agent acting as *Initiator* issues a call for proposals (to perform a particular task) to a set of agents acting as *Participants*. The *Participant* agents have the option to make a proposal or refuse (because they are not interested in the call or have no means to perform the task requested on the call). Once a sensible amount of time has passed (please notice that some *Participants* might not even respond to the call) the *Initiator* gathers the proposals provided and takes a decision. The *Initiator* will either provide a refined call for proposals or consider the set of proposals collected good enough. In the first case, the *Initiator* is effectively returning to the start of the protocol, trying to get better proposals by modifying the call. In the second case, the *Initiator* will accept a set of proposals and reject the remaining ones. The set of *Participant* agents involved in the accepted proposals will perform the associated task and inform about the outcomes.

An agent's negotiation strategy is a specification of the sequence of actions the agent plans to make during negotiation. Such actions will typically consist on offers and responses to offers. There will usually be many strategies that are compatible with a particular protocol, each of which may produce a different outcome [FWJ04]. An agent may change strategy according its utility function and/or scenario.

Thanks to the abstraction level provided by the *ALIVE* model and the FIPA Iterated Contract Net Interaction Protocol (where the central element of design is the role rather than the agent) we can fulfill a two-folded task via the negotiation protocol. On the one hand, the negotiation protocol allows for $\mathcal{I}$ to negotiate with $\mathcal{W}$, in a P2P fashion, in order to reach an agreement for treating industrial wastewaters.

This protocol is initiated by $I_i$, and will occur typically when their wastewater tanks are full, effectively looking for the lower wastewater treatment prize and therefore the maximum benefit, which is the ideal situation from $\mathcal{I}$'s perspective. On the other hand, the protocol allows for $\mathcal{W}$ to negotiate with $\mathcal{I}$ effectively providing offers for wastewater treatment. This scenario will typically occur when plant's demand is *low*, *w.r.t.* plant's treatment capacity, and will allow to any $wwtp_j$ to balance its treatment capabilities, offering to treat wastewater at lower cost when their demand is in valley hours, effectively getting closer to a constant wastewater treatment demand, which is the ideal situation from $\mathcal{W}$ perspective.

## 4.5 Norms

Scenarios like the one introduced in this *Chapter* present several actors (*e.g., Industrial Operator*, *Industrial WWRetainer*, *WWTreater etc.*) with a variety of goals that sometimes are conflicting between them. For instance, the *Industrial Operator* aims at making profit which in turn will generate wastewater, effectively polluting the environment. The *WWTreater* aims at cleaning wastewater for protecting the environment. Therefore, from the individual point of view of an *Industrial Operator*, the more industrial activity the better, even if it results in more polluted water. However, from the individual point of view of a *WWTreater* the less industrial activity the better, as the water will be less polluted and therefore will be easier and cheaper to clean. Bringing this self-interest to the extreme, the ideal situation for *Industrial Operator* is a scenario without environmental protection, where the river can be polluted without constraints. The ideal situation for *WWTreater* is a scenario where there are no polluting elements (no industries, no households) and therefore the river is never polluted. However, the ideal situation from the holistic point of view of the society (as a group of interconnected individuals) is to find a balance to protect the environment while promoting industrial activity.

Furthermore, the roles in the scenario depend on each other for achieving their goals, and therefore they interact in multiple ways. The combination of these two factors results in a society of interacting agents with heterogeneous goals. In order to tame the complexity of these interactions, and to align the overall system with a common high level goal (*e.g.,* protecting the environment without compromising industrial activity) norm-aware electronic distributed systems can be used.

Electronic specifications of norms are one of the mechanisms being applied to define and enforce acceptable behaviour of electronic distributed systems which should comply with some (typically human) regulations. One of the options for providing norm-aware Multi-Agent Systems are Electronic Institutions (*EI*) [VS03]. They are models of human institutions with a norm specification provided in a machine-readable formalism. The main idea behind *EI* is capturing the essence of an institution (mainly norms and protocols) in a machine processable form.

Some functionalities in the system depend on the *Competent Authority* role detecting exceptional situations and applying the appropriate restrictions. For instance, if a meteorological overflow is notified WWPTs can exceed their treatment capacity if they take water from the influent. Therefore plants must limit their influent intake until the overflow is solved. It means water from the influent is not completely treated and to protect the environment industries can not discharge wastewater to the river until the overflow is solved. In order to tackle these restrictions we use *ALIVE*'s normative structure, grounded on regulative and constitutive norms. Such norms specify actor's obligations (WWTPS must limit influent intake until the overflow is solved), prohibitions (discharging water masses with high concentrations of mercury) and permissions (industries can not discharge wastewater to the river until the overflow is solved). In scenarios like the one presented here, Multi-Agent Systems are applied to systems with an overall holistic goal and it is not desirable that an agent's autonomous and emergent behaviour diverges from the overall goal of the system. In order to limit this agent autonomy and ensure a certain coherence between the goals of the particular agents and the overall goals of the system, norms can be applied. Furthermore, norms make the behaviour more predictable, effectively reducing the complexity of the system. Taking this into account, the scenario presented in this *Chapter* provides an exciting new line of research: modelling and implementing the set of norms that will make the system's objectives (*e.g.,* have an environmental sustainable system) stand on top of individuals' objectives (*e.g.,* make profit in the case of the industries). Furthermore, the set of norms provided is not static, as norms will have to evolve through time just as individuals' behaviour changes to adapt to dynamic circumstances. Not only deciding how these norms will have to be adapted is an exciting challenge, but also designing mechanisms to support norm dynamics at run-time. Such mechanisms effectively support adding, removing or updating norms at run-time and while inferring the social state. On the one hand, we can not afford to miss the violation of a norm just because we are updating it. On the other hand, we have to infer a social state consistent with the changes performed in the norms (*e.g.,* it makes no sense to punish an agent for violating a norm that has been removed).

This section provides examples of norms modelled used the framework introduced in *Chapter* 3 and examples on how social and technological changes can affect these norms. For each norm a formal model is provided, as well as the time line depicting the implications of the norm change. Examples for all operations supported by our framework are provided. Norms are inspired on European, national and local wastewater treatment directives.

Our model supports two possible operations (adding and removing norms, accounting norm update as a combination of the two basic operations) in two forms (retroactively and prospectively). The model also supports regulative norms (with obligations, prohibitions and permissions), constitutive norms and institutional powers (including both constitutive powers and normative powers). This accounts for a total of thirty potential examples which are organized as follows:

- Prospectively adding an obligation for $\mathcal{W}$ to treat wastewater before discharging it.
- Retroactively adding an obligation for $\mathcal{W}$ to send water samples to the competent authority.
- Prospectively removing an obligation for $\mathcal{W}$ when unusual situations (*e.g.,* heavy rains) are in place.
- Retroactively removing an obligation for $\mathcal{W}$ to send water samples to the competent authority, due to the deployment of smart sensors over the river.
- Prospectively updating (*i.e.* add and remove) limit concentrations (*i.e.* prohibitions) on industrial wastewaters.
- Retroactively updating (*i.e.* add and remove) the obligation of $\mathcal{I}$ to inform $\mathcal{W}$ of wastewater disposals.
- Prospectively adding a constitutive norm for defining a sensitive area.
- Retroactively adding a constitutive norm for defining a secondary treatment.
- Adding and removing constitutive powers.

*Norm* $N_1$: Let $W_i \in \mathcal{W}$ be a Wastewater Treatment Plant, $M_j \in \mathcal{M}$ a water mass and $T_k \in \mathcal{T}$ a secondary treatment. Once $W_i$ receives a particular water mass $M_j$, the plant has the obligation to treat the water mass with secondary treatment $T_k$ before discharging the water mass.
*Sanction* $S_1$: A generic sanction is applied to the Wastewater Treatment Plant if the norm is not complied with.

| | |
|---|---|
| *Activation Condition* $N_1$ | $received(W_i, M_j)$ |
| *Expiration Condition* $N_1$ | $discharged(W_i, M_j)$ |
| *Maintenance Condition* $N_1$ | $True$ |
| *Deadline* $N_1$ | $performed(T_k, W_i, M_j) \wedge counts\_as(T_k, SecondaryTreatment)$ |
| *Activation Condition* $S_1$ | $isViolated(N_1, W_i)$ |
| *Expiration Condition* $S_1$ | $GenericSanction(W_i)$ |
| *Maintenance Condition* $S_1$ | $True$ |
| *Deadline* $S_1$ | $True$ |

Figure 4.13: Example of formal norm specification for obligation

### 4.5.1 Obligation prospective promulgation:

The European council directive for Wastewater treatment [Cou91] in Article 4 and the Catalan plan for Wastewater treatment inspired on this directive [Gen] state:

> *Member States shall ensure that urban wastewater entering collecting systems shall before discharge be subject to secondary treatment or an equivalent treatment as follows:*
> - *At the latest by 31 December 2000 for all discharges from agglomerations of more than 15 000 p.e. (population equivalent)*
> - *at the latest by 31 December 2005 for all discharges from agglomerations of between 10 000 and 15 000 p.e.*

It means that, by the date '01 January 2006' all Wastewater Treatment Plants (WWTPs) $WWTP_i \in \mathcal{W}$ with a *p.e.* of 10.000 or more have the obligation to perform a secondary treatment (or a treatment that *counts-as* secondary treatment, that is, an equivalent) before discharging water to the river. Failing to comply with the norm will result in the Wastewater Treatment Plant being sanctioned. *Figure* 5.8 shows the formal specification of the regulative norm in our model.

Following the example, the regulative norm is introduced in the system via a Prospective Promulgation operation on the date '01 January 2006'. Therefore, if a particular Wastewater Treatment Plant $W_i \in \mathcal{W}$ with a *p.e.* of 10.000 or more violated the regulative norm (*i.e.* discharged water without treating it) before '01 January 2006', the act has no legal consequences. However, if the plant violates the norm after the promulgation date, it will be sanctioned for the act. In the example depicted in *Figure* 5.15 , a Wastewater Treatment Plant $W_i$ discharges untreated water masses $M_1, M_2$ before norm promulgation without legal consequences. However, discharging untreated water $M_3$ after promulgation results in a sanction being applied.

### 4.5.2 Obligation retroactive promulgation:

The European council directive for Wastewater treatment [Cou91] states in article 15:

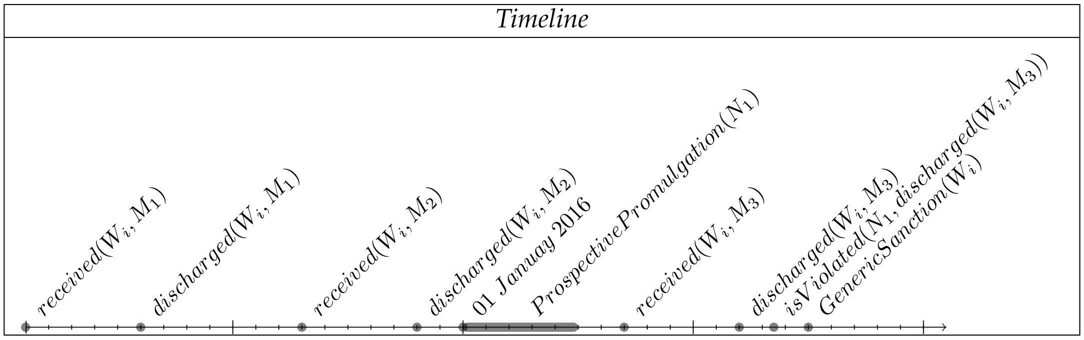



Figure 4.14: Example of timeline for prospective promulgation of a an obligation

*Norm* $N_2$: Let $W_i \in \mathcal{W}$ be a Wastewater Treatment Plant and $S_j \in \mathcal{S}$ the $n^{th}$ water sample taken and sent to the competent authority during the year. Once the year starts, the plant has the obligation to provide the $24^{th}$ sample (*i.e.* $S_{24}$) to the competent authority before the year ends.
*Sanction* $S_2$: A competent authority representative $N_k \in \mathcal{N}$ visits the plant to take water samples

| | |
|---|---|
| *Activation Condition* $N_2$ | $isDay(1) \wedge isMonth(1)$ |
| *Expiration Condition* $N_2$ | $isDay(31) \wedge isMonth(12)$ |
| *Maintenance Condition* $N_2$ | $True$ |
| *Deadline* $N_2$ | $sampleProvided(W_i, S_{24})$ |
| *Activation Condition* $S_2$ | $isViolated(N_2, W_i)$ |
| *Expiration Condition* $S_2$ | $visited(W_i, N_k) \wedge sampleTaken(N_k, S_j)$ |
| *Maintenance Condition* $S_2$ | $True$ |
| *Deadline* $S_2$ | $True$ |

Figure 4.15: Example of formal regulative norm specification for obligation

> *Competent authorities or appropriate bodies shall monitor [...] discharges from urban wastewater treatment plants [...].*

Annex *A.3* of the same document introduces the monitoring conditions, specifically the minimum number of water samples to be taken per year (typically between 12 and 24 per year). In our example, Wastewater Treatment plants have to take the samples (we can set the minimum at 24 for our example) and send them to the competent authority for analysis. Therefore, Wastewater Treatment plants have the obligation to take and send 24 water samples before the end of the year. Failing to comply with the regulative norm will result in the Wastewater Treatment Plant being visited by a competent authority representative. The objective of the visit is to take water samples (*i.e.* repair action). *Figure* 5.9 shows the formal specification of the norm in our model.

Following the example in *Figure* 5.16, the regulative norm is introduced in the system via a Retroactive Promulgation operation. This allows to classify efficiently the plants that require an inspector to take samples. If a particular plant (*e.g.*, ($W_1$ in the example) has

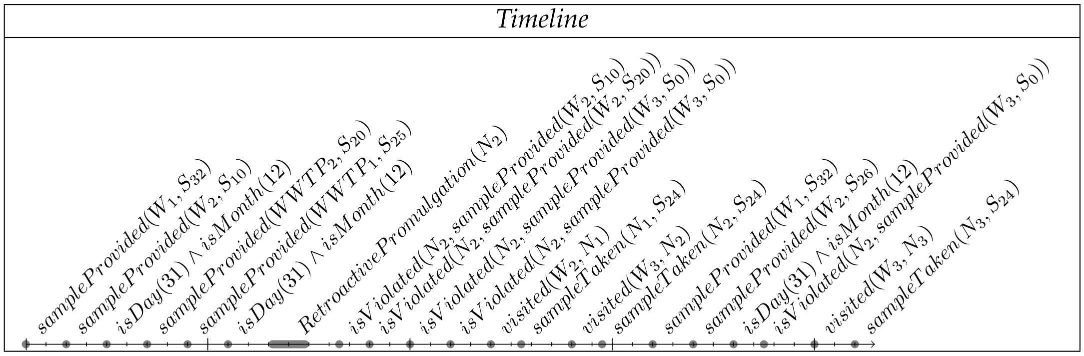



Figure 4.16: Example of timeline for retroactive promulgation of an obligation

always taken and provided the minimum samples required (even if the norm was not in place, for instance, because the minimum was just a recommendation) it can avoid the official inspection. However, if a plant has not reached the minimum (*e.g.,* $(W_2)$ or has never provided a sample in the past (*e.g.,* $(W_3)$ it will be visited by a competent authority representative to perform an inspection taking water samples. Therefore, when norm $N_2$ is promulgated, plants requiring an inspection are effectively classified, because past violations of the regulative norm are detected. Needless to say, once the norm is in place, plants that do not provide the samples are sanctioned (*i.e.* inspected). Please note how retroactive promulgation allows to check agent's behaviour $w.r.t.$ the norms in the past, effectively providing a method to identify agents inclined to abide with them.

### 4.5.3 Obligation abrogation:

The European council directive for Wastewater treatment [Cou91] states in *Annex I* point *D.5*:

> *Extreme values for the water quality in question shall not be taken into consideration when they are the result of unusual situations such as those due to heavy rain.*

In our example it stands for the obligation to perform a secondary treatment not having effect in unusual situations, such as heavy rain. The obligation has already been introduced in §4.5.1 and formally modelled in *Figure* 5.8. However, the model does not take into account the fact that the norm is not in place in case of unusual situations. One option is to include the exception on the model of the norm (*i.e.* preventing the norm from activating if water is received by the plant, but an unusual situation is in place). Formally, it would imply substituting Norm's $N_1$ activating condition, which is currently $received(W_i, M_j)$ for $received(W_i, M_j) \wedge \neg unusualSituation()$. However, this solution would result in more complex norm formalizations. Furthermore, if new exceptions to the norm are added, more conditions would be included in the activating condition, resulting in complex and hard to understand norms. A cleaner solution is to allow the competent authority (or any other actor with power to alter the norms that govern the system) to temporally remove the norm from the system when it is considered appropriate (in our example, while the unusual situation takes place). On the one hand, these norm could be used to keep norms simple and easy to understand. That is because we are leaving the decision of which

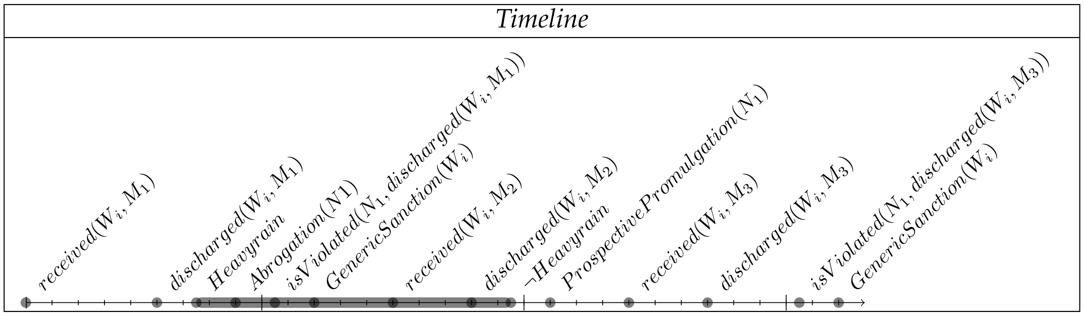



Figure 4.17: Example of timeline for the abrogation of an obligation

norms should be active in every scenario to higher level (and more expressive) reasoning processes performed by the agents responsible of introducing and removing norms in the system. On the other hand, all the exceptions to the different norms do not have to be taken into consideration at norm design time. They can be introduced later when designing a process that decides which norms are active in the system at every point of time. That is, our approach allows supporting a Normative System which is truly dynamic and adapts to changing (and sometimes even not foreseen) situations.

Following the example depicted in *Figure* 5.17, the norm $N_1$ is removed from the system via an Abrogation operation. This allows to effectively implement a general exception to the norm while an unusual situation of heavy rain takes place. Therefore, if a particular Wastewater Treatment Plant $W_i \in \mathcal{W}$ violated the norm (*i.e.* discharged water without treating it) in a situation of heavy rain, the act has no legal consequences. However, if the plant violates the norm outside the unusual situation, it will be sanctioned for the act. In this example, a Wastewater Treatment Plant $W_i$ discharges untreated water masses $M_2$ during heavy rain without legal consequences. However, discharging untreated water $M_1, M_3$ outside the unusual situation results in a sanction being applied. Please note that one of the sanctions (associated to the discharge of $M_1$) is applied during the unusual situation. This is because the action causing the norm violation occurred outside the unusual situation, and our framework is expressive enough to detect this particular fact.

### 4.5.4 Obligation annulment:

The European council directive for Wastewater treatment [Cou91] states in article 15:

> *Competent authorities or appropriate bodies shall monitor [...] discharges from urban wastewater treatment plants [...].*

This obligation has already been introduced in §4.5.2 and formally modelled in *Figure* 5.9 via the obligation to provide samples from Wastewater Treatment Plants to the corresponding competent authority. However, this method to monitor discharges from Wastewater Treatment Plants is not very reliable. Mainly because it depends on the good will of the plants taking and providing the samples. This is the main reason why regulative norms are required to promote such necessary behaviour, and repair actions are applied in case the norms are not complied with (*i.e.* inspections for plants not providing enough samples per year). Technological advances, such as the Internet of Things [Kop11] may provide

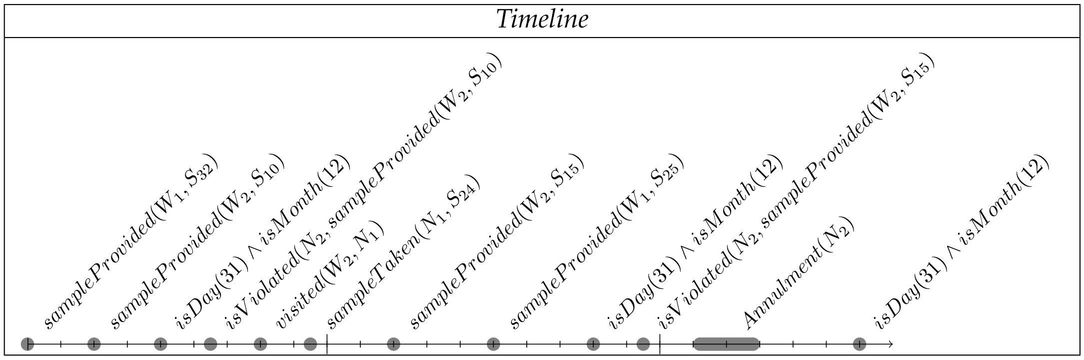



Figure 4.18: Example of timeline for annulment of an obligation

better methods to perform these monitoring processes. Connecting smart objects (*i.e.* embedded systems such as sensors) to the Internet makes it possible to remotely access sensor data, effectively providing real time monitoring capabilities to the areas covered by such sensors. Furthermore, the monitoring capabilities are provided with low latency (several water samples can be taken per hour) and continuously (as sensors can operate 24 hours per day and 7 days per week). One of the main ideas behind our scenario is deploying smart sensors over the river basin. Therefore Wastewater Treatment Plants do not have to provide water samples anymore in order to monitor the state of the river. Furthermore, Wastewater Treatment Plants which did not provide water samples in the past will not require an inspection anymore, as the data can be accurately provided by the sensors.

Following the example presented in *Figure* 5.18, when smart sensors are deployed along the river basin, the norm $N_2$ is removed from the system via an Annulment operation. Therefore, Wastewater Treatment Plants no longer have the obligation to send water samples to the competent authority. Furthermore, if water samples were not provided in the past, plants will not have to go through an inspection. Mainly because information provided by smart sensor readings is equivalent to the information provided by an inspector agent. This allows to effectively cancel pending inspections because both water monitoring and inspection is performed now by a network of smart sensors. Once the norm $N_2$ is abrogated, plants do not have to take and provide samples. Furthermore, if a particular plant (*e.g.,* $WWTP_2$ in the example) has neither taken nor provided the minimum samples required in the past, it can avoid the official inspection. This is because abrogation will not only remove the norm from the system, but also the associated sanctions.

### 4.5.5 Prohibition prospective update:

European, national and regional directives impose a limit on pollutant concentrations of water discharged to the different ecosystems (*e.g.,* river, sea, river basin, *etc.*). An example of such limits is the one in [Gen] *Section* 4.5.2, where a table with pollutant limits is provided, as depicted in *Figure* 4.19. The table covers pollutants such as iron, soap, oils and fats, mercury, *etc.* For instance, the mercury limit is fixed at $0.005\ mg/l\ Hg$. *Figure* 5.12 shows the formal specification of the normative limit for mercury in our model.

| PARÀMETRE | UNITATS | LÍMIT D'EMISSIÓ SEGONS MEDI RECEPTOR | | |
|---|---|---|---|---|
| | | LLERA (1) | MAR (2) | SISTEMA (3 |
| CLORURS | mg/l Cl | 2.000 | – | 2.500 |
| ALUMINI | mg/l Al | 1 | 10 | 20 |
| FERRO | mg/l Fe | 2 | – | 10 |
| SULFURS | mg/l S | 1 | – | 1 |
| SULFATS | mg/l $SO_4$ | 2.000 | – | 1.000 |
| OLIS I GREIXOS | mg/l | 20 | – | 250 |
| DETERGENTS | mg/l LSS | 2 | 12 | 6 |
| DETERGENTS NO IÒNICS | mg/l NP-10 | (2) | 8 | 1 |
| CADMI | mg/l Cd | 0,1 | 0,4 | 0,5 |
| MERCURI | mg/l Hg | 0,05 | 0,1 | 0,1 |
| COURE | mg/l Cu | 0,2 | 4 | 3 |
| CROM TOTAL | mg/l Cr | (2) | 3 | 3,0 |
| NÍQUEL | mg/l Ni | 2 | 5 | 5 |
| PLOM | mg/l Pb | 0,2 | 2 | 1 |
| CIANURS | mg/l CN | 0,5 | 5 | 1 |
| FLUORURS | mg/l F | 6 | – | 12 |
| FOSFATS | mg/l $PO_4$ | (30) | 90 | – |
| AMONI | mg/l $NH_4$ | 16 | – | 60 |
| NITROGEN AMONIACAL | mg/l N | 12 | – | 47 |
| NITRITS | – | – | – | – |
| NITRATS | mg/l $NO_3$ | 44 | – | 100 |
| AOX | mg/l | (0,5) | 0,5 | 2 |
| TRICLOROETILÈ | µg/l TRI | 200 | 200 | 400 |
| PERCLOROETILÈ | µg/l PER | 200 | 200 | 400 |
| SUMA TRICLOROBENZÈ | µg/l | 100 | 100 | 200 |
| HEXACLOROBENZÈ | µg/l | 2.000 | 2.000 | – |
| SUMA HEXACLOROCICLOHEXÀ | µg/l | 4.000 | 4.000 | – |

(1) Els valors límit de llera s'han adoptat de:

- a) Taula 3 de l'annex al títol IV del Reglament del domini públic hidràulic (RD 849/1986)
- b) Límits màxims (en altres seccions són inferiors) vigents d'abocament a llera pública per a substàncies de la llista I: Directiva 76/464/CEE i derivades de desenvolupament (82/176/CEE, 83/513/CEE, 84/156/CEE, 84/491/CEE, 86/280/CEE, 88/347/CEE, 90/415/CEE i 91/692/CEE), com també les seves transposicions per les ordres 12-11-87, 13-3-89, 27-2-91 i 28-6-91.
- c )Els valors que figuren entre parèntesis són estimats a l'efecte operatiu de càlcul.

(2) Per als abocaments a mar, s'han tingut en compte els límits d'emissió inclosos en l'annex IV del Conveni de Col·laboració entre l'extinta Junta de Sanejament (actual Agència Catalana de l'Aigua) i l'AEQT (Associació Empresarial Química de Tarragona) que data de l'any 1999, complementats amb els límits màxims vigents d'abocament per a substàncies de la llista I.(id. 1b)

(3) Pel que fa als abocaments a sistema, els límits corresponen al Decret 130/2003 de 13 de maig pel qual s'aprova el Reglament dels serveis públics de sanejament.

Figure 4.19: Table with normative limits for pollutants as seen in directive PSARU2005 [Gen].

As new scientific studies are performed new pollutants might be included in the list (*e.g.*, emerging pollutants such as hormones that can be harmful for river flora and fauna) or limits for actual pollutants updated. Therefore, if regulative norms taking into account such pollutant limits are promulgated, they are likely to change over time. In our framework we support updating a norm by removing the old version of the norm first, and then

*Norm* $N_3$: Let $W_i \in \mathcal{W}$ be a Wastewater Treatment Plant, $M_j \in \mathcal{M}$ a water mass and $concentration(M_j, Hg)$ the concentration of mercury in the water mass. It is forbidden for Wastewater Treatment Plants to discharge water masses to the river with a mercury concentration higher than $0.005$ $mg/l$. Please note that the norm is always active, as the prohibition always holds, therefore activation condition is set to $True$ and expiration condition to $False$.
*Sanction* $S_3$: A generic sanction is applied to the Wastewater Treatment Plant if the norm is not complied with.

| | |
|---|---|
| *Activation Condition* $N_3$ | $True$ |
| *Expiration Condition* $N_3$ | $False$ |
| *Maintenance Condition* $N_3$ | $discharged(W_i, M_j) \wedge concentration(M_j, Hg) \leqslant 0.005mg/l$ |
| *Deadline* $N_3$ | |
| *Activation Condition* $S_3$ | $isViolated(N_3, W_i)$ |
| *Expiration Condition* $S_3$ | $GenericSanction(W_i)$ |
| *Maintenance Condition* $S_3$ | $True$ |
| *Deadline* $S_3$ | $True$ |

Figure 4.20: Example of formal norm specification for prohibition

inserting the new version of the norm back into the system.

Following the example in *Figure* 5.19, let us suppose the mercury limit is updated from $0.005$ $mg/l$ $Hg$ to $0.0025$ $mg/l$ $Hg$ as this pollutant is likely to affect human beings via the food chain. Updating the regulative norm will imply Abrogating norm $N_3$ as defined in *Figure* 5.12 and prospectively promulgating norm $N_3'$. Please note that norm $N_3'$ is similar to norm $N_3$ but with the new limit in the *Maintenance Condition* of the norm. It is relevant to note that one of the sanctions (associated to the discharge of water mass $M_2$ by plant $W_1$) is applied after regulative norm Abrogation. This is because the action causing the norm violation occurred before the norm was removed from the system (due to the update operation), and our framework is powerful enough to detect this particular fact.

### 4.5.6 Prohibition retroactive update:

In our scenario, industries are connected to Wastewater Treatment Plants for treating industrial wastewater before discharging it to the different ecosystems (usually, a river). In order to control plant treatment capacity, identify its needs and foresee plant updates, industries interact with plants via an operational contract. The contract specifies the obligation of the industry to communicate incoming discharges to the associated plants, along with discharge characteristics. At the same time, the contract sets a limit on the amount of water and pollutant concentration a particular industry can discharge to a particular plant. Thanks to these contracts, the plants can effectively plan their water treatment capacities at real-time, adapting them to industrial demand. At the same time, experts from the competent authority can evaluate plant water treatment capacity, as they know the maximum water and pollutant concentration load a particular plant could have at any moment in time. This is because they know the industries connected to a particular plant, and industries have limited the maximum amount of water and pollutants they can send to the plant for treatment, via the contract. *Figure* 5.13 shows the formal specification of the contract (via regulative norms) in our model. In this case, the contract contains both an obliga-

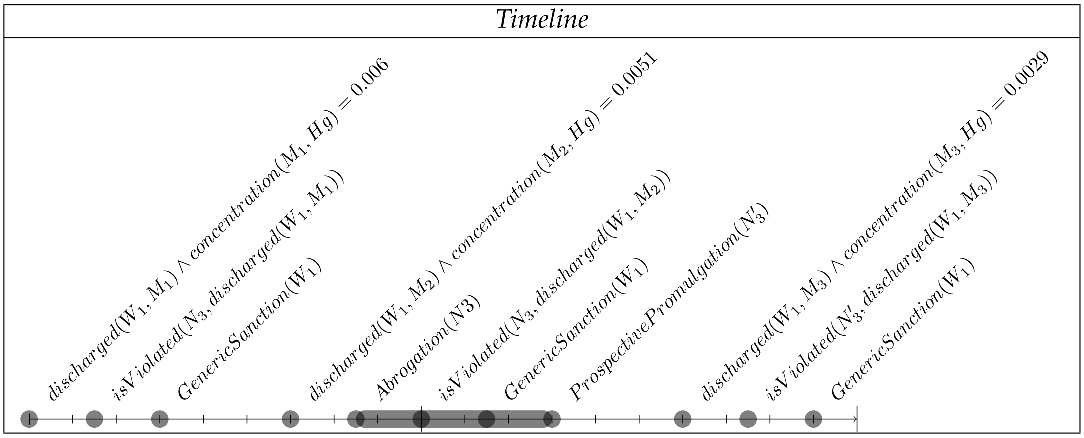



Figure 4.21: Example of timeline for prospective update of an obligation

tion (inform before discharging water to the plant) and a prohibition (at any moment in time, never discharge a higher pollutant concentration that the amount specified by the contract), and both are modelled in the same norm.

Similar to the example provided in §4.5.4, when technological advances are introduced (*e.g.*, smart sensors are included in the river), the norm becomes obsolete and is updated. In this case, including smart sensors in the river excludes the obligation to inform about discharges from the contract. Furthermore, competent authority experts decide that sanctions incurred for not informing in the past should be abolished, as this act of goodwill with industries will promote the introduction of smart sensors in the river basin. However, sanctions associated to overpassing pollutant concentrations limits should remain. As in §4.5.5 updating the norm will imply removing norm $N_4$ (as defined in 5.13) and adding a new norm $N_4'$ which is just like $N_4$ but with the *Deadline* set to $True$. As competent authority experts want to remove sanctions associated to not informing about discharges, the norm $N_4$ should be abrogated. It means, all industries sanctioned in the past for not informing about discharges will receive a Generic compensation. However, as the norm is abrogated as a whole (rather than abrogating only the *Deadline*) industries sanctioned for overpassing the limits will receive the compensation as well. This can be solved by adding the new norm $N_4'$ retroactively, as it will check for limit overpassing instances in the past (neglecting missing informs) and sanction them, effectively annulling the compensations provided for overpassing pollutant limits. Please note that the expressibility of our framework allows to update norms with retroactive effects (just like in the example provided in this subsection) or without them (just like in the example provided in §4.5.4). Furthermore, two more operations are possible:

- The old norm is removed retroactively (*i.e.* annulled) and the new one added prospectively.
- The old norm is removed prospectively (*i.e.* abrogated) and the new one added retroactively.

In the example provided in *Figure* 5.20 industry $I_1$ is sanctioned for not informing about

*Norm* $N_4$: Let $W_i \in \mathcal{W}$ be a Wastewater Treatment Plant, $M_j \in \mathcal{M}$ a water mass, $O_k \in \mathcal{O}$ a pollutant, $concentration(M_j, O_k)$ the concentration of a pollutant in the water mass, $I_l \in \mathcal{I}$ an industry and $maxConcentration(W_i, I_l, O_k)$ the maximum concentration of a particular pollutant (as allowed by the contract between industry $I_l$ and plant $W_i$). It is forbidden for industries to discharge water masses to a Wastewater Treatment Plant with a pollutant concentration higher than the one specified in the contract. Furthermore, industries have the obligation to inform the Wastewater Treatment Plant before discharging the water. Please note that the norm expires when the water mass is discharged by the industry, and immediately activates again (as long as there is a contract between the industry and the plant) ready for the next discharge.
*Sanction* $S_4$: A generic sanction is applied to the Industry if the contract is not complied with.

| | |
|---|---|
| *Activation Condition* $N_4$ | $SignedContract((W_i, I_l)$ |
| *Expiration Condition* $N_4$ | $discharged(I_l, M_j)$ |
| *Maintenance Condition* $N_4$ | $concentration(M_j, O_k) \leqslant maxConcentration(W_i, I_l, O_k) \forall O_k \in \mathcal{O}$ |
| *Deadline* $N_4$ | $InformDischarge(I_l, W_i, M_j)$ |
| *Activation Condition* $S_4$ | $isViolated(N_4, I_l)$ |
| *Expiration Condition* $S_4$ | $GenericSanction(I_l)$ |
| *Maintenance Condition* $S_4$ | $True$ |
| *Deadline* $S_4$ | $True$ |

Figure 4.22: Example of formal contract specification with obligations and prohibitions

a discharge and industry $I_2$ is sanctioned for overpassing pollutant limits. When the norm is updated, both industries receive a generic compensation on norm $N_4$ removal. However, industry $Ind_2$ looses the compensation as it is sanctioned again when the new norm $N_4'$ comes in place. For simplicity we focus on a single pollutant concentration $Hg$ that has a contractual limit of $0.005\ mg/l\ Hg$.

### 4.5.7 Constitutive prospective promulgation:

The European council directive for Wastewater treatment [Cou91] states in *Article* 5 points 1 and 2:

1. *For the purposes of paragraph 2, Member States shall by 31 December 1993 identify sensitive areas according to the criteria laid down in Annex II.*
2. *Member States shall ensure that urban wastewater entering collecting systems shall before discharge into sensitive areas be subject to more stringent treatment than that described in Article 4, by 31 December 1998 at the latest for all discharges from agglomerations of more than 10.000 p.e.*

Please note that *Article* 5 provides an exception for *Article* 4, which is introduced in §4.5.1 and formally modelled in *Figure* 5.8. *Article* 5 states that Wastewater Treatment Plants with a p.e. of 10.000 or more should perform before discharge a more stringent treatment. This means a more stringent treatment than a secondary treatment (or equivalent). The norm applies as long as the plants are located in sensitive areas. Sensitive areas are identified by member states (*i.e.* competent authorities). The concept of sensitive area is an institutional concept (*i.e.* it has meaning in the scope of an institution) that classifies

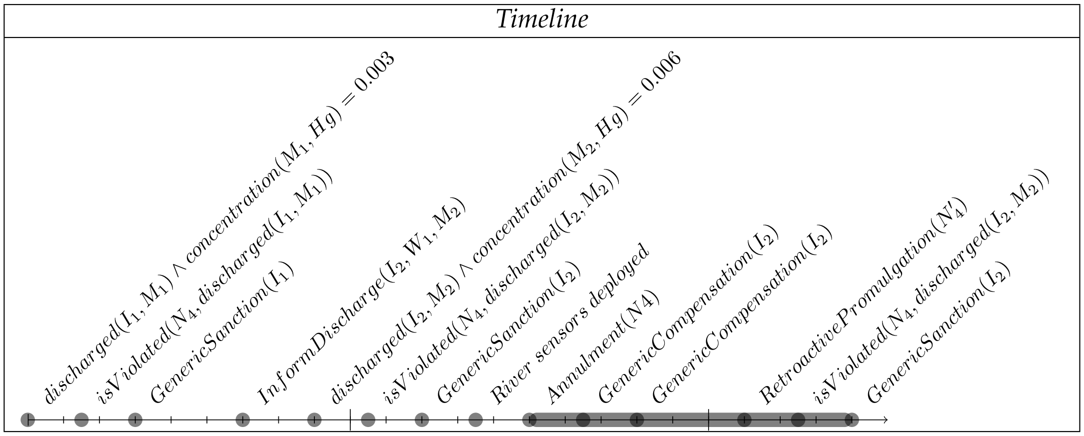



Figure 4.23: Example of timeline for retroactive update of an obligation

*Norm* $N_5$: Let $W_i \in \mathcal{W}$ be a Wastewater Treatment Plant with a p.e. of 10.000 or more, $M_j \in \mathcal{M}$ a water mass, $T_k \in \mathcal{T}$ a stringent treatment and $location(W_i)$ the location of plant $W_i$. Once $W_i$ receives a particular water mass $M_j$, the plant has the obligation to treat the water mass with a stringent treatment $T_k$ before discharging the water mass, as long as the location of the plant is a sensitive area.
*Sanction* $S_5$: A generic sanction is applied to the Wastewater Treatment Plant if the norm is not complied with.

| | |
|---|---|
| *Activation Condition* $N_5$ | $received(W_i, M_j) \wedge counts\_as(location(W_i), SensitiveArea)$ |
| *Expiration Condition* $N_5$ | $discharged(W_i, M_j)$ |
| *Maintenance Condition* $N_5$ | $True$ |
| *Deadline* $N_5$ | $performed(T_k, W_i, M_j) \wedge counts\_as(T_k, StringentTreatment)$ |
| *Activation Condition* $S_5$ | $isViolated(N_5, W_i)$ |
| *Expiration Condition* $S_5$ | $GenericSanction(W_i)$ |
| *Maintenance Condition* $S_5$ | $True$ |
| *Deadline* $S_5$ | $True$ |

Figure 4.24: Example of formal norm specification for obligation with constitutive norm

special areas as *sensitive area*. Formally, it means a particular area *counts-as sensitive areas* in the scope of an institution. Therefore, sensitive areas can be introduced in our normative context via constitutive norms, supported by the $counts\_as$ construct. The formal model of the norm inspired by the article is depicted in *Figure* 5.14.

Following the example depicted in *Figure* 4.25, a particular location $L_1$ is not considered a sensitive area. Therefore, if a Wastewater Treatment Plant $W_i \in \mathcal{W}$ with a *p.e.* $\geq$ 10.000 and located in $L_1$ discharges water performing just a secondary treatment, the act has no legal consequences. This is because norm $N_1$ is complied with and norm $N_5$ does not activate. However, if the plant discharges water performing just a secondary treatment after the location is declared a sensitive area, it will be sanctioned for the act. In the example a Wastewater Treatment Plant $W_i$ discharges water masses $M_1, M_2$ without legal consequences. However, discharging water $M_3$ after the area is declared sensitive results in a sanction being applied.

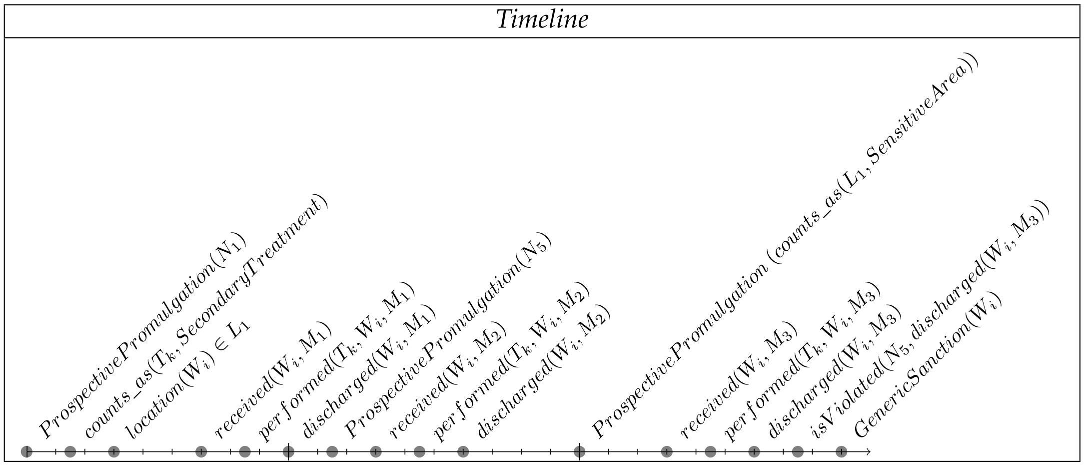



Figure 4.25: Example of timeline for prospective promulgation of a constitutive norm

Please note that an analogous example can be constructed for prospectively removing constitutive norms. Following the example depicted in *Figure* 4.26, a particular location $L_1$ is considered a sensitive area. Therefore, if a Wastewater Treatment Plant $W_i \in \mathcal{W}$ with a *p.e.* $\geq$ 10.000 and located in $Loc_1$ discharges water performing just a secondary treatment, the act results in a sanction being applied. However, if the statement declaring the area sensitive is removed the plant can discharge water performing just a secondary treatment with no legal consequences. In the example a Wastewater Treatment Plant $W_i$ discharges water masses $M_1, M_2$ being sanctioned for this. However, after the statement declaring the area sensitive is removed, the plant discharges water $M_3$ with no legal consequences.

### 4.5.8 Constitutive retroactive promulgation:

The European Council directive for Wastewater treatment [Cou91] in Article 4 and the Catalan plan for Wastewater treatment inspired on this directive [Gen] state:

> *Member States shall ensure that urban wastewater entering collecting systems shall before discharge be subject to secondary treatment or an equivalent treatment as follows:*
>
> - *At the latest by 31 December 2000 for all discharges from agglomerations of more than 15 000 p.e.*
> - *at the latest by 31 December 2005 for all discharges from agglomerations of between 10 000 and 15 000 p.e.*

*Article* 4 has already been introduced in §4.5.1 and formally modelled in *Figure* 5.8. As new scientific studies are performed new water treatment methods are developed and plants may deploy them. At a particular point in time, an institutional actor (*e.g.*, competent authority) may state a treatment is equivalent to a secondary treatment. Formally,

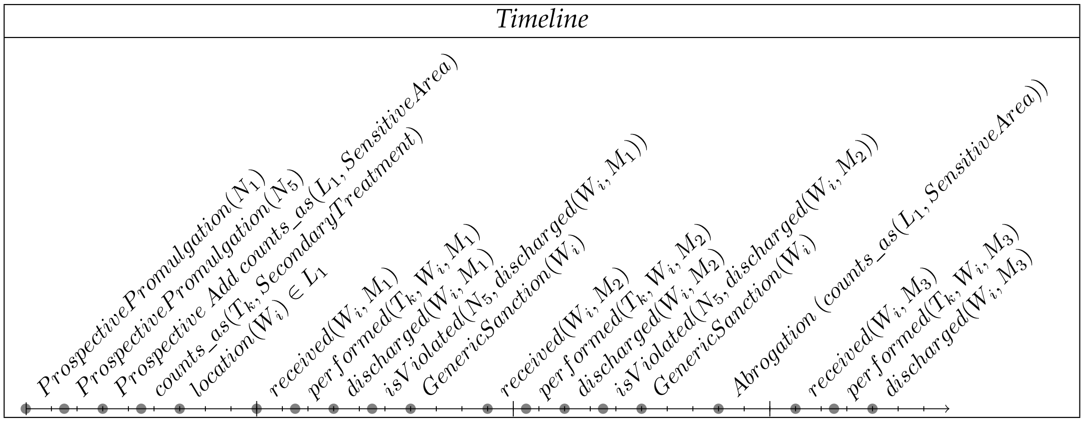



Figure 4.26: Example of timeline for abrogation of a constitutive norm

a particular new treatment *counts-as* secondary treatment (*i.e.* a regulative norm equivalent to the construct $counts\_as(T_k, SecondaryTreatment)$). And this statement should be taken into account with retroactive consequences. If a new treatment is equivalent to a secondary treatment at this point of time, it was equivalent in the past (we assume the new treatment has not changed). Following the example depicted in *Figure* 4.27, a Wastewater Treatment Plant $W_i \in \mathcal{W}$ is applying an innovating treatment $T_k$ that uses nano-bots to capture contaminant particles in the water. As treatment $T_k$ has no institutional equivalence to a secondary treatment $W_i$ is sanctioned for treating water masses $M_1, M_2$ with it. Once the treatment $T_k$ has recognised institutional equivalence to a secondary treatment (*i.e.* $counts\_as(T_k, SecondaryTreatment)$) $W_i$ can treat water masses (*e.g.*, $M_3$) with the new treatment with no legal consequences. Furthermore, as $T_k$ is considered to be institutionally equivalent to a secondary treatment also in the past (from the beginning of time) $W_i$ receives a couple of compensations for both sanctions entailed from treating water masses $M_1, M_2$ with $T_k$.

Please note that an analogous example can be constructed for retroactively removing constitutive norms. Following the example depicted in *Figure* 4.28, a Wastewater Treatment Plant $W_i \in \mathcal{W}$ is applying a particular treatment $T_k \in \mathcal{T}$ based on Navajo dreamcatchers to clean the water. As treatment $T_k$ has an institutional equivalence to a secondary treatment (*i.e.* $counts\_as(T_k, SecondaryTreatment)$) $W_i$ is not sanctioned for treating water masses $M_1, M_2$ with it. Once scientific studies demonstrate $T_k$ is based on a pseudoscience, the constitutive norm is removed and $T_k$ looses its recognised institutional equivalence to a secondary treatment. Therefore, if $W_i$ treats water masses (*e.g.*, $M_3$) with the treatment $T_k$ it will be sanctioned for not using a secondary treatment. Furthermore, as $T_k$ is considered to be an invalid treatment also in the past (from the beginning of time) $W_i$ is sanctioned treating water masses $M_1, M_2$ with $T_k$, even if such water masses were treated before the constitutive norm was removed.

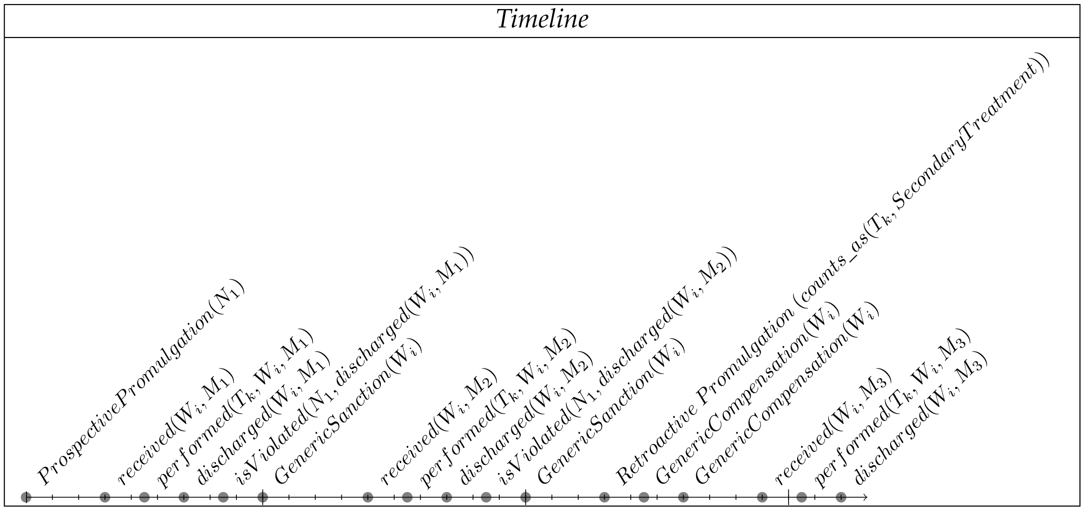



Figure 4.27: Example of timeline for retroactive promulgation of a constitutive norm

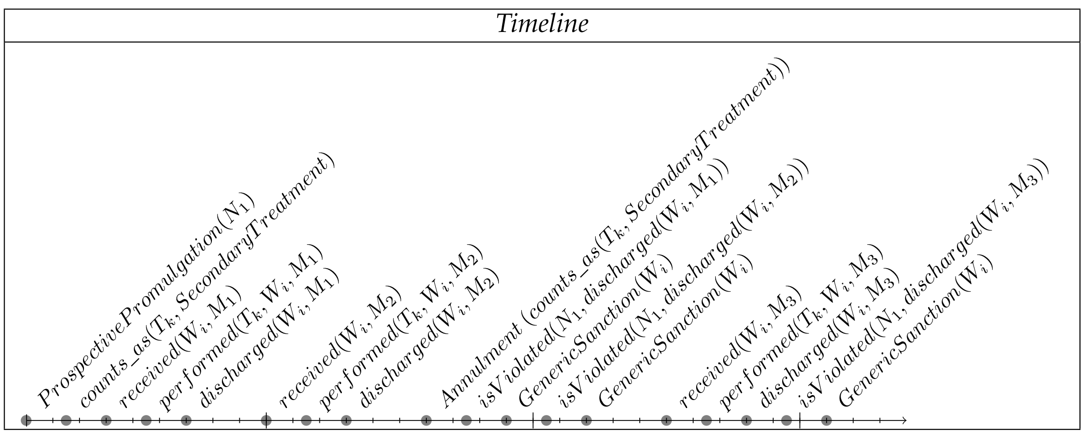



Figure 4.28: Example of timeline for annulment of a constitutive norm

### 4.5.9 Constitutive power

The European council directive for Wastewater treatment [Cou91] states in *Article* 5 point 1:

> *For the purposes of paragraph 2, Member States shall by 31 December 1993 identify sensitive areas according to the criteria laid down in Annex II.*

In the case of Spain, identification of sensitive areas is delegated on regional governments since December 1995 as seen in point 7.3 of [Gov]. Before December 1995 regional governments in Spain (*e.g.*, Catalan government) lacked constitutive power to declare sensitive areas. Therefore, the declaration of a sensitive area from a regional government had no legal consequences, whereas the same declaration from the national government had full legal consequences.

The example depicted in *Figure* 4.29 is inspired in the norm introduced in §4.5.7 and formally introduced in *Figure* 5.14. The norm $N_7$ introduces the obligation of Wastewater Treatment Plants to perform a more stringent treatment if they are located in sensitive areas (*i.e.* discharging to sensitive areas). Following the example, a particular plant $W_i \in \mathcal{W}$ is not sanctioned for discharging water masses (both $M_1$ and $M_2$) without performing more stringent treatment. That is, the plant is just performing a secondary treatment $T_1$. This is because the declaration of sensitive area from the local government has no legal consequences, and therefore norm $N_5$ is not activated (and norm $N_1$ is complied with). Once the regional government has constitutive power to declare sensitive areas, the plant violates norm $N_5$ and is sanctioned for discharging water masses (*e.g.*, $M_3$) without performing a more stringent treatment.

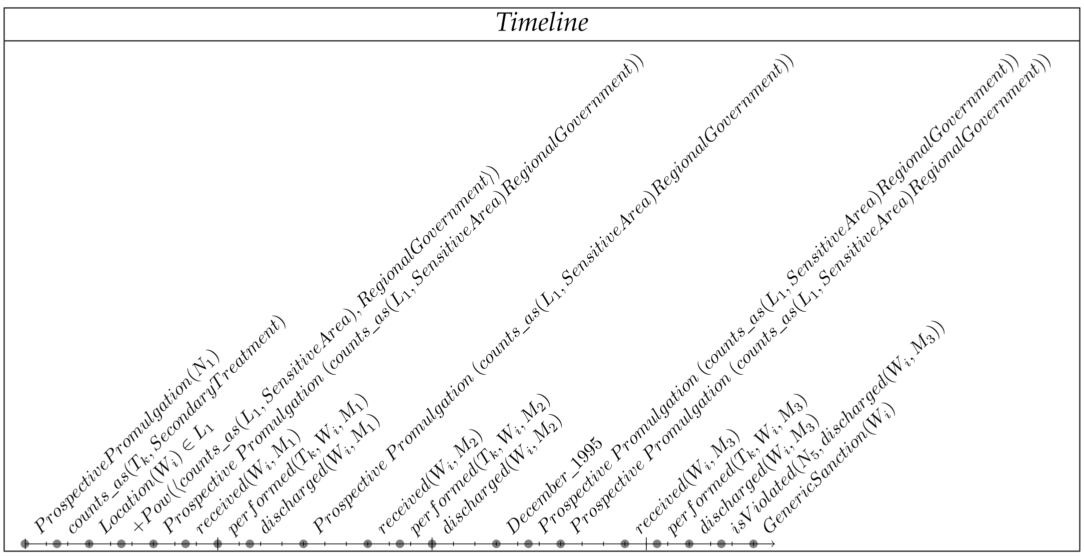



Figure 4.29: Example of timeline for prospective promulgation of a constitutive power

An analogous example can be constructed for removing the power prospectively. For instance, in the example provided, when power is given to the regional governments it is withdrawn from the national government. It will invalidate institutional statements uttered by the national government (*w.r.t.* sensitive areas) since power removal. Following the example depicted in *Figure* 4.30 plant $W_i \in \mathcal{W}$ is not sanctioned for discharging water masses (both $M_1$ and $M_1$) without performing a more stringent treatment. This is because the national government has lost its constitutive power to declare sensitive areas and therefore the declaration of sensitive area has no legal consequences. As a result norm $N_5$ is not activated (and norm $N_1$ is complied with) and the plant is not sanctioned.

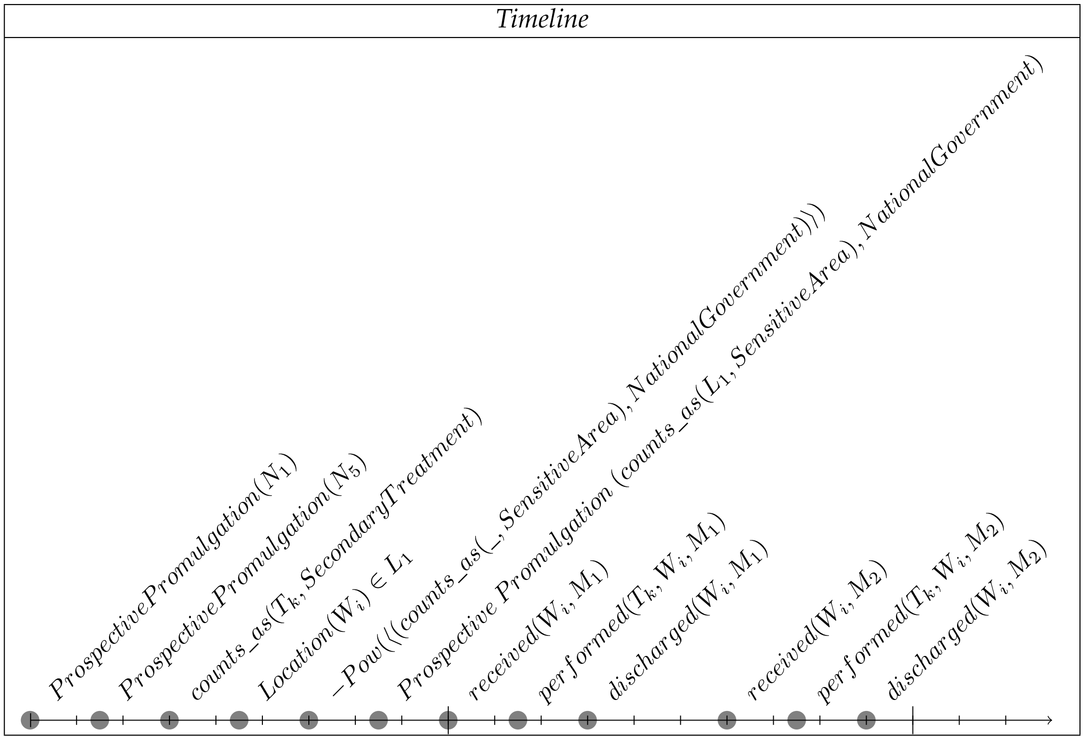



Figure 4.30: Example of timeline for abrogation of a constitutive power

## 4.6 Scenario implementation

Once we have introduced our theoretical framework for the testing scenario we proceed to depict the implementation performed for our testing framework. We will focus on depicting the architecture and the set of norms resulting from simulating the different cases in our testing scenario.

As depicted in *Figure* 4.31 we have performed the implementation by simulating the behaviour of the different actors involved in the river basin scenario in an agent platform. We have used the JADE agent platform [BPR01] because it allows to easily develop and deploy agents based on the Java programming language. The platform is FIPA-compliant supporting negotiation protocols such as contract-net protocol [Smi80] that can be applied to the scenario presented in this document. The agents in the platform perform tasks associated to the actors in the scenario they simulate. The tasks are passed through the event bus to the NoMoDEI platform where the monitor puts them in contrast with the set of norms in the scenario. A static program triggers norm changes that have prospective or retroactive impact on the normative state of the system. Finally, we use a norm visualizer to check the impact of agent actions and norm operations in the scenario.

*Figures* 4.32 and 4.33 show the annulment of a norm in our test scenario. The example corresponds to *Norm* 5.9 and time-line 5.16 in §4.5. *Figures* 4.34 and 4.35 show the update

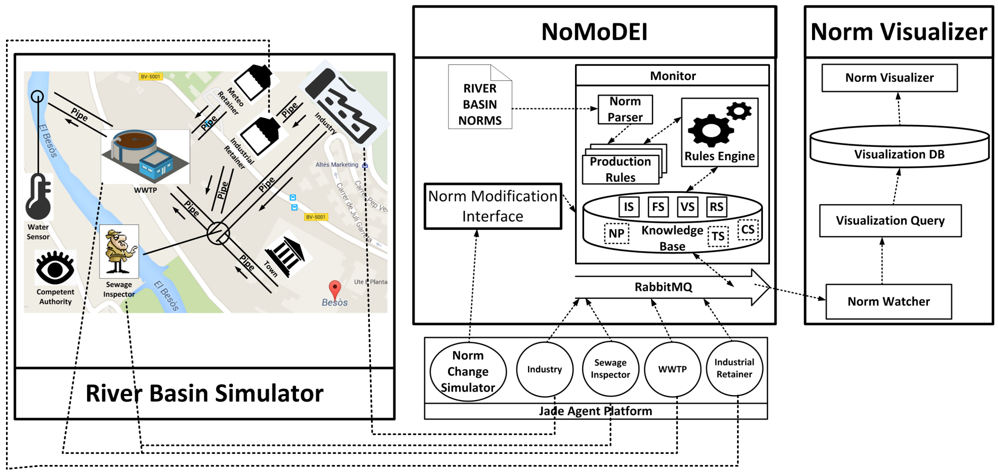



Figure 4.31: Architecture for our testing scenario

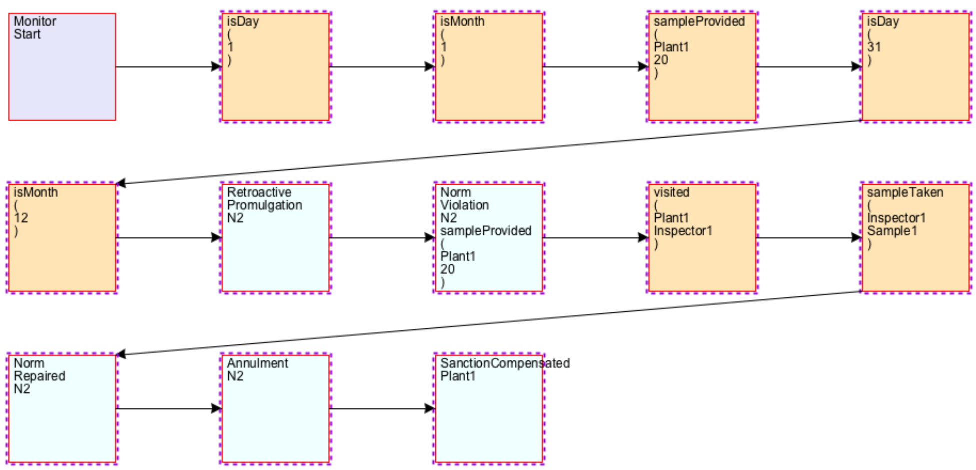



Figure 4.32: Annulment of a norm: timeline

of a norm in our test scenario. The example corresponds to *Norm* 5.13 and time-line 5.20 in §4.5. Finally, *figures* 4.36 and 4.37 show the effect of a constitutive power in our test scenario.

Via our testing implementation we have demonstrated we can integrate regulations and protocols in an agent simulation, bridging the gap between simulations and social simulations by embedding social constructs (*i.e.* norms) in the simulation platform. Agent interactions are observed and put in contrast with the regulations regimenting the scenario, aligning agent's actions with a common high level goal shared by the society. Fur-


Figure 4.33: Annulment of a norm: norm events

thermore, regulations and protocols are allowed to evolve through time, reflecting changes in the society as well as exceptional situations that may cause the norms to cease being in force.

## 4.7 Conclusions and Related work

Once we have finished introducing our approach of a norm-aware agent-based model for integrated Wastewater management systems, we proceed to put it in contrast with similar approaches. This section compares the state of the art analysed in §2.3 with the work presented in this *Chapter*. We will provide a brief summary of every proposal analysed in §2.3 and compare it to our proposal, focusing on detecting confluence points where our

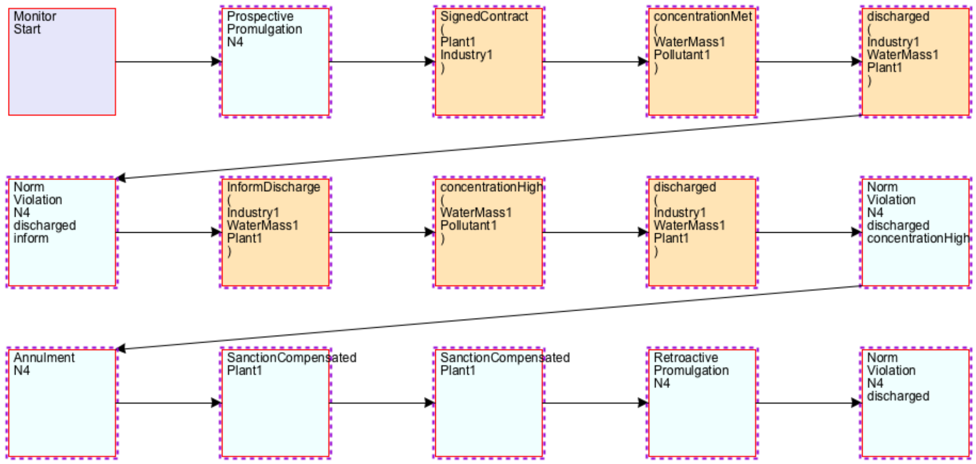



Figure 4.34: Update of a norm: timeline

work can be complemented by the different approaches in the state of the art. Then, we will outline conclusions.

### 4.7.1 Distributive Justice for Self-Organised Common-Pool Resource Management

In §2.3.1 we analyse an approach for enduring common pool resource management. Even though the approach is generic and not directly applied to water management, one can easily see both water and water treatment capabilities (*e.g.,* public wastewater treatment plants) can be represented as a common pool of resources. Therefore, the approach is directly applicable to water management. For instance, the incentive to use a water resource before someone else leads to pumping races in water management scenarios, which has been identified as a case of the tragedy of the commons. Ostrom [Ost90] demonstrates that such tragedy can be effectively avoided and cites many cases where the common resource is sustained over generations [LGC06]. In these cases, the actors (*i.e.* appropriators) formed an institution, defining a set of rules that regulate and constraint resource provision, effectively self-governing the commons.

The proposal presented in §2.3.1 combines Ostrom's institutional design principles [Ost90] with Rescher's theory of distributive justice [Res66] based on the concept of legitimate claims. The idea is presenting an approach for enduring common pool resource management, where a common pool of resources is consumed by agents in a cluster. Agents will leave the cluster if they consider resource distribution is not fair, according to their individual point of view.

The proposal is able to ensure fairness in demand and provisioning, effectively providing an enduring cluster where no actor is encouraged to leave due to a non-fair distribution of resources. However, appropriation presents a challenge. Actors can cheat on appropriation (effectively bypassing allocation) resulting in an unfair distribution of resources. Authors claim that, in order to tackle this issue, they need methods for preventing this behaviour, such as using retributive justice to punish cheating actors. Authors propose to


Figure 4.35: Update of a norm: norm events

use monitoring and sanctioning mechanisms to identify non-compliant behaviours (*e.g.*, appropriating more resources than the ones allocated) and fine cheating actors.

When it comes to applying monitoring and sanctioning mechanisms to identify non-compliant behaviours and fine cheating actors the proposal presented in §2.3.1 and our proposal meet. Their approach is similar to the one we present, where actors can violate the rules of the game as long as they consider it is beneficial from an individual or even social perspective. It is system designer's responsibility to define a set of sanctions such that non-compliant behaviours are de-promoted, and only in exceptional situations agents will consider it is beneficial to violate norms. Furthermore, our proposal supports adapting regulations defining compliant behaviour at run-time. We consider the approach presented in §2.3.1 could benefit from our proposal, where rules of resource appropriation

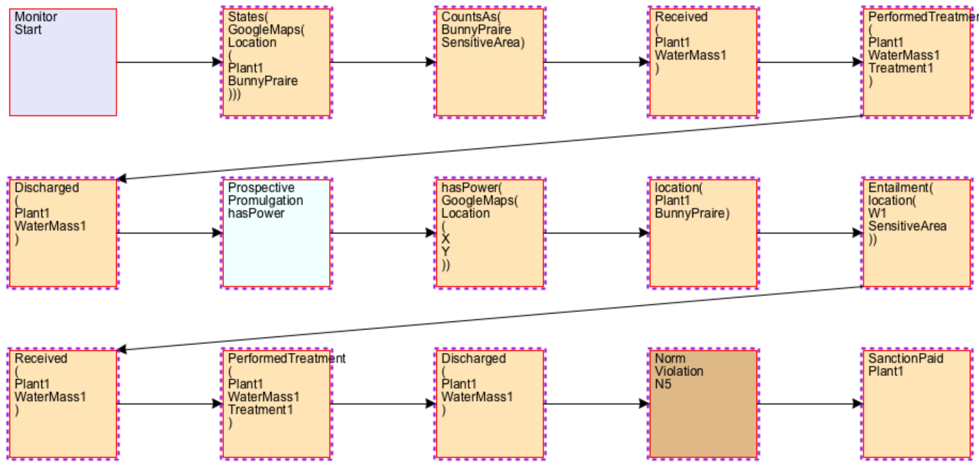



Figure 4.36: Constitutive power: timeline

can be adapted to a wide range of contextual situations, such as the state of the pool of resources (*e.g.*, applying different regulations to full pools than the ones applied to nearly depleted ones), the state of the actors involved (*e.g.*, taking into account the frequency of regulation violation or agent cluster size) or even environmental situations (*e.g.*, heavy rains or droughts that impact the regeneration of the common pool of resources).

### 4.7.2 Improving urban wastewater management through an auction-based management of discharges

In §2.3.2 we analyse an auction-based coordination process for industrial discharges to WasteWater Treament Plants in a way plant's capacity (for treating water quantity or pollutant concentration in the water) is never exceeded. During the coordination process proposed, every industry agent will communicate the WWTP agent its discharge schedule for a given period of time (*e.g.*, a day). The schedule contains the discharges planned for a given period of time, and for every discharge information about water quantity and pollutant concentration. Once the WWTP agent has received all the schedules for a given day it starts checking for conflicts, where the plant exceeds its capacity. When a conflict is detected. the involved industry agents are informed about it, and an auction is started to solve it, effectively forcing some industries to modify their discharge schedule. In case it has to reschedule, the industry agent will try to store the rejected discharge into its retention tank. The discharge of the tank is scheduled as the first action to be performed by the agent once the conflict has been solved. Please note that if tank is already full (or there is no tank) the discharge will be performed anyway. Authors propose a mechanism [MMB+07] to minimize these situations during the auction process, but to the best of our knowledge, it is not applied. Furthermore, authors assume industries are naive and they will not try to cheat. They state in the real world application a mechanism to ensure the bid price corresponds to the industry's urgency for discharging should be implemented.

The coordination process proposed shows the benefits of applying high level institu-

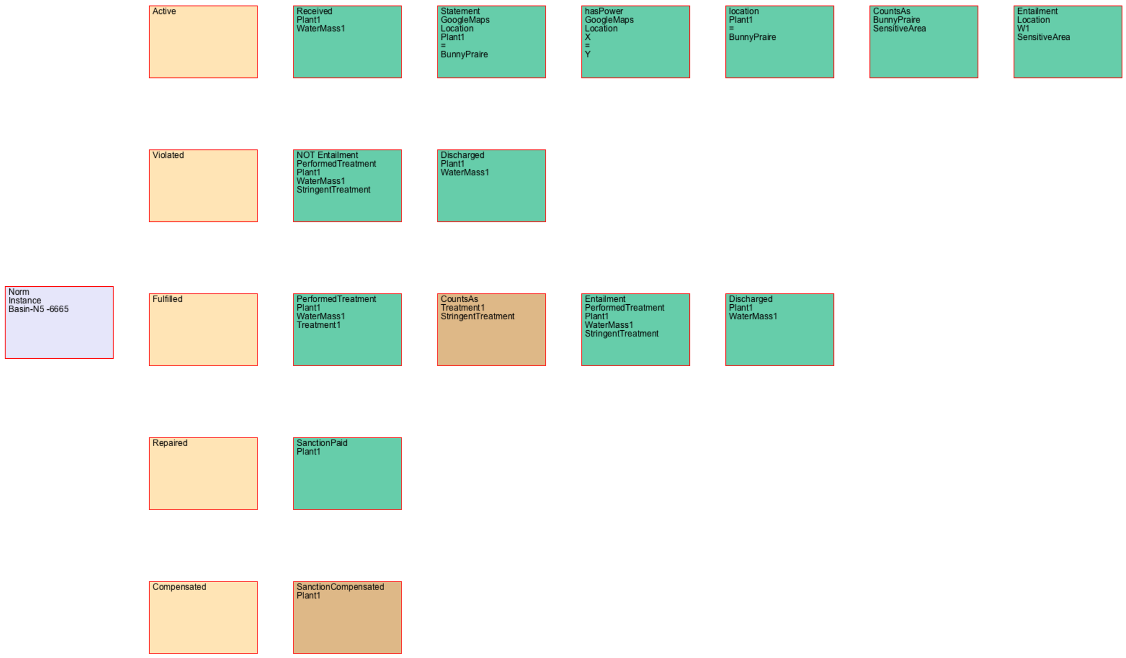



Figure 4.37: Constitutive power: norm events

tional definitions to water management scenarios. Compared to our approach, the proposal in §2.3.2 is more specific (*e.g.,* it is applied only in conflict detection) and also less expressive. Our proposal consists in a continuous negotiation between plants and industries (*e.g.,* every time the industry wants to discharge, it will negotiate with the plant) which allows not only to avoid exceeding plant's capacity, but also to avoid periods of inactivity, where plant's treatment capacities are wasted. This is because setting a prize for water treatment and negotiating will encourage industries to make use of valley hours, where the plant is idle, and therefore the cost for water treatment can be reduced. Our proposal can even support situations where the plant is treating water for free (or even paying industries for treating it) in the event it needs an industrial wastewater discharge with some particular characteristics to continue its operation.

The mechanisms to detect and avoid cheating agents proposed in §2.3.2 are more simple that the ones we propose. On the one hand we complement cheater detection with regulations and protocols proposed by a competent authority that mirrors the regulations and protocols we can find in real world scenarios. On the other hand we do not aim at detecting industries lying on their water treatment requirements (we consider such behaviours will be minimized or even disappear in the long run), but detecting defecting actors in one of the following situations:

- Industries sending more water than the amount agreed during the negotiation process, or with different characteristics. A special case of this scenario is sending water to a plant without a previous negotiation.

- Industries that exceed their maximum quotas for water quantity or pollutant concentration. Please note, this allows to avoid plant overload in a flexible way. If the plant is idle and other industries have confirmed they do not plan to treat wastewater in the near future, a particular industry quota could be exceeded if the involved industry pays higher treatment costs.
- Treatment plants that understand the water quality parameters they have imposed via regulations and protocols. They are able to understand the effects of such parameters and in which exceptional cases they might be bypassed.
- Defecting actors taking advantage of exceptional situations. Industries discharging wastewater in heavy rain situations, when their wastewater parameters can not be verified. Industries sending wastewater when the plant is not operating properly (droughts plant maintenance operations) or when the plant is busy treating exceptional pollution from the river or from a particular industry (*e.g.*, caused by accidents such as a broken retention tank).

Summarizing, we consider our proposal is complementary with the one presented in §2.3.2. We could benefit from implementing a more elaborated negotiation processes that applies only when negotiation is required (applying pre-defined water treatment costs otherwise) and the proposal in §2.3.2 could benefit from applying more flexible and open negotiation protocols. At the same time, the proposal in §2.3.2 could benefit from our Normative System to better detect cheating agents and enforce compliant behaviour. The option to make agents understand regulations and protocols could also be beneficial for the proposal presented in §2.3.2, as the negotiation process can be clearly affected by the regulations and protocols in place.

### 4.7.3 Ant Colony Optimization-based Method for Managing IndustrialInfluents in Wastewater Systems

In §2.3.3 we analyse a system for coordinating industrial wastewater discharges based on ant colony optimization. The system aims at finding the best combination of industrial discharges *w.r.t.* WWTP efficiency, that is, as much capacity as possible is used from the plant without overloading it. For doing it, ants are randomly placed on a graph-like search space, where nodes are industrial activities and edges possible discharges.

The work neglects some issues such as the efficiency of a centralized decision system in real-world scenarios, or the need to have complete information about industry production plans in order to entail expected industrial wastewater discharges. Please note that in real world scenarios industries might not be willing to share this information, specially when competing industries are involved in the discharge coordination process. Furthermore, the work presented in §2.3.3 does not take into account agents not abiding to the expected patterns of behaviour, and lacks methods to deal with such issues. The clearest case is industries discharging different amounts of water or with different pollutant concentrations than the ones scheduled by the coordinating agents. Finally, the work does not cover the case where a particular industry lacks retention tanks, and therefore, its discharge must be treated, as there is no option to delay it.

The work in §2.3.3 focuses on the internal reasoning process of a coordinator agent (wich can be fulfilled by the WWTP agent in our scenario) whereas our approach focuses on the structure of the agent society and the interactions among agents from an organizational point of view, without detailing the internal reasoning processes of the agents. Therefore, we could state both approaches are complementary and can benefit from each other. When put in contrast with our approach, where industry agents negotiate with plants before discharging, the work presented in in §2.3.3 shows potential for being integrated into a WWTP agent reasoning process. The WWTP agent could analyse industry proposals (taking into account they are the expected discharge) look for conflicts, where the WWTP capacity is exceeded, and solve them using the ant colony based optimization process. This would allow for deciding which discharges to accept from industries and establishing a base cost for the negotiation, while keeping industrial production plans private. Furthermore, the work in §2.3.3 could benefit from the Normative System we propose for enforcing acceptable patterns of behaviour (*e.g.,* industries comply with the agreements they reach with WWTPs for wastewater treatment) as well as from our institutional model to facilitate coordination in complex scenarios where a particular agent can fulfil more than one role (*e.g.,* industries with their own WWTP that can accept other industry wastewater as long as they are paid enough for treating it).

### 4.7.4 Integration of freshwater environmental policies and wastewater treatment plant management

In §2.3.4 we overview a simulation based analysis of the different directives for regulating the characteristics of the discharged water and the chemical characteristics of the received ecosystems.

The work runs several simulations based on the model of a WWTP in the north-east of Spain, to conclude it is possible to integrate the chemical status of receiving water bodies into WWTP management, effectively adjusting plant operation conditions to the pollution loads measured not only in the influent, but also in the river. It recommends updating the legislation to account for an integrated perspective allowing for a more flexible management of the WWTP.

At first glance, it seems the work presented in §2.3.4 is far away from our proposal. However, taking into account we are proposing to integrate Normative Systems (modelling real-world regulations and protocols) into wastewater treatment scenarios we find a clear connection point: we can use our proposal to support a Multi-Agent Systems that will simulate the effect of environmental policies just like in §2.3.4. Furthermore, as our proposal supports a normative context that can evolve over time, we can support policies that can be adapted to different situations, such as:

- Season of the year. Taking into account different water temperatures and rainfall parameters. Furthermore, we do not need to run different simulations modelling different seasons like in §2.3.4, we can perform season change (adopting a different set of policies) during the simulation, effectively representing the transition period between seasons.
- Exceptional situations including heavy rains, droughts, maintenance periods on WWTPs and river pollution caused by accidents in industries. Such exceptional situations will require putting in place different sets of policies and regulations to solve the contingency. Our proposal supports the change, so we can effectively model it.
- Agent behaviour. When regulations are too restrictive, too limp or do not take into account the whole scenario (as shown in §2.3.4) agents might find them unfair. As in our proposal, agents have the option to violate regulations we might find a particular unfair regulation is being violated frequently. In such situations we can effectively detect the behaviour, analyse it and update the set of regulations regimenting the system, effectively updating unfair ones with better regulations. This approach is

specially promising when running simulations to test an improve a set of regulations (*w.r.t.* a particular distribution of agent behaviours) and in norm emergence [UM15] scenarios.

### 4.7.5 $m$Water, a Case Study for Modeling Virtual Markets

In §2.3.5 we analyse $m$Water, a regulated virtual market simulation where autonomous agents trade rights for the use of water in a closed basin.

The idea behind $m$Water is allowing policy makers to compare different market configurations using market performance indicators. Market configurations contain the following parameters;

- Simulation start dates and duration.
- Participant population, supporting different behavioural templates.
- Regulations and protocols to be used during the negotiation process.
- Key decision points to regulate the decisions taken by the participants. In the implementation analysed such decision points rely on a random basis but there are plans to extend them with techniques based on short-term planning, trust, argumentation and even ethical values.

According to the authors, such simulation tool provides the following advantages:

- Includes a model for concepts on water regulation, water institutions and individual behaviour of water users.
- Represents the interactions between regulations, institutions and individuals.
- Emphasizes actor's participation in decision making.
- Provides a tool to evaluate modifications on regulations and protocols before applying them to the real world.

Just like our approach, $m$Water aims at narrowing the gap between water management simulations (based on equational descriptions) and social simulations. The motivation behind social simulations is to mimic the behaviour of autonomous rational individuals and groups of individuals [SHWS09]. The main idea is modelling not only hydraulic factors (which can be perfectly modelled using equational systems) but also social factors, including:

- Norm typology. Including the set of regulations and protocols governing the system.
- Actor's behaviour (and misbehaviour). Including mechanisms (based, for instance, in Normative Systems) to detect non-compliant behaviour and enforce a compliant one.
- Trust and reputation criteria. Effectively supporting long-term collaboration agreements supported by institutional settings.
- Actor willingness to cooperate to achieve common higher level goals. Promoted by the factors presented above in the list.

$m$Water and our proposal have several characteristics in common. Both are social simulations grounded on Electronic Institutions able to represent roles, coordination scenes, objectives and a Normative System. However, when compared to our approach $m$Water presents a more specific and in-depth proposal. $m$Water focuses on negotiation for water use rights, whereas our proposal covers the whole river basin management scenario, therefore the negotiation process is not presented with such detail. Furthermore, while $m$Water correctly emphasises the need to flexible and dynamic Normative Systems (*e.g.,* authors stress the need of *'organization schemes that are flexible and able to adapt to a changing* *environment with multiple situations'*) no method for supporting them is presented. Our proposal clearly remarks that this method is available, and we provided an exhaustive set of examples based on river basin management. On the one hand, we consider our proposal could benefit from the work done in $m$Water for implementing more expressive and powerful auction mechanisms when negotiating for wastewater treatment resources. On the other hand, we consider $m$Water could benefit from our proposal to widen the application scenario (limited not only to interactions involved on the negotiation for water use rights, but covering the whole set of interactions present in river basin management) and support dynamic Normative Systems, able to change the set of norms during the simulations, effectively adapting them to new situations and requirements. This would allow to simulate not only new sets of policies but also sets of policies evolving through time, allowing to evaluate not only the impact of the new set of policies in the system, but also the performance of the evolution (*e.g.,* measuring how long does it take for the new policies to be adopted and the performance of the system during the transition between different sets of policies). In general we consider evaluating the impact on the system of policy evolution, while the simulation keeps running and the different actors pursuing their objectives, opens new, more realistic and exciting lines of future work *w.r.t.* simulations for policy optimisation.

### 4.7.6 Conclusions

This *Chapter* has presented a norm-aware agent-based model for integrated wastewater management systems. The *Chapter* provides an example on how Normative Systems can be integrated in Multi-Agent Systems where actors' objectives are heterogeneous and sometimes conflicting. The Normative System allows to align agent's objectives with common organisational objectives. At the same time, it allows to detect undesirable patterns of behaviour in the agents, such as free raiders. Thanks to our proposal, misbehaving actors can be sanctioned, effectively enforcing good practices among the actors.

In this aspect, our proposal shows many features in common with several works in the state of the art. However, our proposal goes beyond, as it allows the set of norms governing the Multi-Agent System to evolve through time. We provide a wide range of examples, where regulative norms in the form of obligations, prohibitions and permissions are inserted, removed and updated. Furthermore, we also show examples of dynamic operations on constitutive norms and constitutive powers.

While most of the systems analysed show a less expressive normative language (they typically do not account for constitutive norms and constitutive powers) we provide a rich set of normative elements, supporting deontic elements (obligations, prohibitions and permissions), constitutive norms, constitutive powers and violation handling norms (*i.e.* sanctions). Furthermore, our normative elements contain a rich structure with activation, maintenance and deactivation conditions, as well as deadlines.

Finally, we support norm dynamics, which is not supported by the proposals analysed in the state of the art. We propose four operations to update the Normative System accounting for norm promulgation and derogation both in prospective and retroactive forms. On the one hand, we combine norm operations with a rich set of normative elements providing a dynamic normative language that can be adapted to a numerous set of contexts and situations. This is specially important in wastewater management scenarios, where the set of norms will evolve adapting to situations which are typically out of control of managers and legislators (*e.g.,* heavy rains, droughts, pollution of the environment). On the other hand we can adapt norms while our system is on-line, inferring a normative state consistent with the update. In scenarios like wastewater management we can not afford to stop observing the social reality, as free raiders and other misbehaving actors could take advantage of this situation.

In contrast, our proposal does not present complex reasoning processes and decision taking mechanisms for the agents involved in the system. We focus on the Normative System, so we can effectively benefit from more expressive and complex agents the other proposals include.

In this chapter, we have seen a wide range of norms and norm operations, inspired in real world regulations and protocols. The chapter focuses on how the Normative System can evolve. On the one hand adapting to new regulations and protocols caused by technological advances. On the other hand adapting to unexpected situations which are typically out of control of managers and legislators, such as heavy rains.

As a summary, the main contributions of this chapter are:

1. A model of the agents involved in wastewater management in the river basin.
    a) The model includes a social structure developed using the ALIVE [APV$^+$10] methodology.
    b) The model includes a normative structure specified in the norm formalism we use on this document.
2. An instantiation of the generic architecture presented in §3 to the wastewater management scenario presented in this *Chapter* .
3. Implementation details and tests on the architecture.
4. The proposal in this *Chapter* establishes the basis for performing agent-based social-aware simulations in the river basin scenario.

The river management scenario introduced in this *Chapter* has been developed in collaboration with Mr. Luis Oliva and Doctors Ulises Cortés, Manel Poch and Marta Verdaguer.

Once we have applied our proposal to wastewater management scenarios we proceed to next *Chapter* where we apply it to e-health systems. The next *Chapter* will focus on medical regulations and protocols that specify good medical practices. Our system will be used to:

- Detect and solve in a timely manner potentially dangerous situations, such as a patient not taking his medication.
- Control security issues related to medical data, such as an unauthorized doctor accessing a patient medical record.
- Control medical protocols, such as preventing self-medication.

CHAPTER 5

# A Practical Use case: Dynamic Electronic Institutions to support patient treatment adherence

Population ageing, defined as a process which increases the proportion of old people within the total population, is one of the main problems of this century as older population (aged 60 years or over) is estimated to grow from the current 11% to 22% by 2050 [Pop12]. The size of the population aged between 65 and 80 + years in Europe (EU-27) today is 80 million senior citizens, with a doubling of this figure forecasted by 2050 [Sch08]. Population ageing affects or will affect both developed and developing countries. Moreover, the cost of supporting an elder is greater than the cost of supporting a child in a ratio of five to three [Uni04], most of this cost being caused by higher health expenses.

The epidemiological shift in disease burden from acute to chronic diseases over the past 50 years has rendered acute care models of health service delivery inadequate to address the health needs of the population. Chronic diseases are a major and growing problem in the population. They are by far the leading cause of mortality, representing some 86% of all deaths in the EU. Adherence to a treatment was defined by the WHO's adherence project [HO03] as the extent to which a person's behaviour – taking medication, following a diet, and/or executing lifestyle changes, corresponds with agreed recommendations from a health care provider. Adherence to long-term therapy for chronic illnesses in developed countries averages 50%. In developing countries, the rates are even lower. It is undeniable that many patients experience difficulty in following treatment recommendations. Poor adherence to long-term therapies severely compromises the effectiveness of treatment making this a critical issue in population health both from the perspective of quality of life and of health economics. Interventions aimed at improving adherence would provide a significant positive return on investment through primary prevention (of risk factors) and secondary prevention of adverse health outcomes [HO03].

In the coming years this situation (together with other economic factors) will put great pressure on the national healthcare budgets, mainly because therapies for managing chronic diseases (*e.g.*, hypertension, diabetes, depression, Parkinson, *etc*) are performed away from the institutional care setting, typically at home. This distributed approach to daily care requires that patients, specially elders, be capable of autonomously taking several different medications at different time intervals over extended periods of time. This can easily lead to forgetfulness or confusion when following the prescribed treatment, specially when the patient is suffering multiple pathologies that require a treatment with a cocktail of drugs. This gets worsened when elders suffer a cognitive impairment. Medication compliance is a critical component in the success of any medical treatment.

Initiatives attempting to address medicine non-adherence promote patient involvement in treatment decisions but remain ineffective in older patients or in patients with cognitive disorders. Interventions using applied high-technology show potential for supporting medication adherence in patients with diseases that requires poly-pharmacological treatment. Both concordance and adherence management are of high priority, having a significant effect on the cost effectiveness of therapy. This is especially important where there are disorders with a high healthcare costs, such as oncological diseases, psychiatric disorders, HIV, geriatrician disorders or dementia. Pharmaceutical care could help to reach optimal cooperation between patients, the healthcare professional using high technology interventions, as one of the main objectives is to improve the rate of patient adherence in long-term therapy. Adherence to therapy can vary from 0% to 65% depending on the client population and the type of treatment. Adherence rates are typically higher in patients with acute conditions, as compared to those with chronic conditions, with adherence dropping most dramatically after the first six months of therapy and in prophylaxis [Nat07]. Patients' non-adherence to a therapeutic regimen may result in negative outcomes for them and may be compounded in populations with multiple morbidities that require multiple drug therapy. The elderly exemplifies such population. Adherence may also be affected by access to medications, which may be restricted by the use of formularies or insurance programmes. However, non-adherence may represent a greater risk in older people resulting in poor disease control that may be compounded with multiple morbidity and polypharmacy. There are many reasons why patients do not follow their therapy as prescribed. Maybe they find that they cannot tolerate the side effects. It may be that the high cost of some medicines prohibits acquisition of their medication until such time that they have been able to accumulate enough resource to purchase their repeat prescription. Where a condition is asymptomatic, such as Hypertension, the patient may be lulled into thinking that their treatment has worked and that they no longer require to take their medication or follow their diet; distracted by the hectic pace of everyday life, perhaps they simply forget to take their pills.

Whatever the reason, treatment non-adherence is an expensive and potentially deadly problem, resulting in 89,000 deaths and 100 billion US dollars per year in unnecessary hospital costs, in the USA. It has been estimated that there are 194,500 deaths a year in the EU due to miss-dose and non-adherence of prescribed medication. Non-adherence is estimated to cost the European Union 125B Euro annually. Unused medications returned in the UK are incinerated. In UK alone 10% of all drug waste is incinerated each year about, 369.6M Euro.

Information technologies applied to health care show potential for supporting medication adherence, specially in patients with diseases that require poly-pharmacological treatment. By using such technologies, pharmaceutical care could help to reach optimal cooperation between patients and healthcare professionals. This will effectively improve the rate of patient adherence in long-term therapy and provide a more fluent relationship between patients and healthcare professionals.

Due to the distributed, multi-institional nature of pharmacological treatment prescription and execution, An agent-based solution [WJ95, Wei99a] can better model the different actors, along with their needs and responsibilities. Agent-based systems can be applied to senior and disabled patients, effectively increasing their ability to live an independent life. The agents will provide presonalised assistance in carrying out activities of daily living and health care maintenance, such as treatment adherence. At the same time, they will facilitate both communication and interaction with the different actors involved in healthcare (ranging from patients and relatives to doctors and pharmacists). These tools can be effectively used to assist the health care and social interaction of patients, delaying their institutionalisation by prolonging the period of relative independence [CMA01].

The increasing dependence on information technologies in health care organisations has increased the interest in security techniques applied to healthcare. Typically, security is concerned with the protection of information from unauthorized access, either while stored or communicated. It is widely accepted that sensitive medical data (ranging from complex genome information to simple medical records) must be dealt with special care regarding security. It is clear that research on secure access to data will be fundamental in ensuring any software component in general and agents in particular may access or update sensitive information. However, in highly regulated scenarios involving several tasks that must be coordinated by a range of actors, both the roles for some of the actors (*e.g.,* who can act as patient's caregiver) and the way they should or may interact with patients (*e.g.,* who can access patient compliance records) are clearly defined and regulated, and this requires security from a higher level point of view [FD00]. Not only sensitive data must be protected, but the activities carried out by the actors should be controlled, and this implies that the rules and protocols regimenting the medical organisation where the agents operate must be represented, understood by the different agents (either human agents or computational processes) and enforced[Nor90]. Including such rules and protocols into the system will not only enhance security but also social acceptance (from patient's point of view) and professional acceptance (from doctor's point of view).

Therefore, the need to use information technologies that comply with pre-defined patterns of behaviour (*i.e.* medical regulations and protocols) arises. Electronic specifications of norms are one of the mechanisms being applied to define and enforce acceptable behaviour of electronic distributed systems which should comply with some (typically human) regulations. But such regulations and the environments they are applied to are not static. Taking this into account, norm-aware information systems should be capable of adapting to normative change. Changes in normative environments are typically caused by environmental changes, social changes or technological changes. An example of an environmental change is a patient that travels through the European Union along the year, which requires a information system designed to accommodate (sometimes conflicting) national and international regulations, policies and protocols governing patient's (pharmacological) treatment, taking at all times into consideration the current location of the patient into account. An example of social change is the death of a patient's relative that was playing the role of personal caregiver at home. This will require to redistribute the responsibilities to other relatives, close friends or institutional actors to ensure that all tasks and responsibilities are covered. A clear example of a technological change is the implementation of e-Prescription systems. As such systems imply a connection, from an information perspective, of the different actors involved with the medication (*e.g.,* doctor, pharmacist and patient) they will typically redistribute responsibilities among actors. For instance, via e-Prescription systems patients do not have the obligation to go themselves (or send a relative) to the pharmacy to present a physical prescription, as a copy of the e-Prescription is sent to the patient's pharmacy of choice. Furthermore, e-Prescription may support pharmacies in sending the medication to the patient directly.

However, regulations and protocols are subject to social and technological changes that make them obsolete, effectively requiring their replacement with new regulations and protocols. As an example, the introduction of the e-Prescription in many countries has updated the medical protocols followed to distribute medication on pharmacies. It also the case that some information systems (*e.g.,* systems designed to support treatment adherence on a patient that travels through the European Union along the year) arise the necessity to accommodate (sometimes conflicting) national and international regulations, legislations and protocols governing patient's treatment.

In this *Chapter* we present an extension of the COAALAS project (COmpanion for Ambient Assisted Living on *ALIVE*-SHARE-*it* platforms) [GSGGÁN11] applied to a m-health scenario in the scope of the *AVICENA* project. We enrich the framework with a dynamic Normative System grounded on the ALIVE framework [ÁNCPVS09] that allows us to create a computational model for the regulations and protocols around treatment prescription and treatment adherence.

The COAALAS project is a framework for multi-agent systems that combines organisational and normative theories with Ambient Assisted Living (AAL) technologies. The *AVICENA* project proposes the development of an innovative m-Health platform and well-tailored personalized services to improve chronic patients' medication and treatment adherence. *AVICENA* includes several modules (*e.g.,* a smart pill dispenser, a social network, *etc.*) coordinated as a society via the COAALAS framework. The system we introduce provides three main properties:

- Flexibility: Rather than restricting the actions of the different actors involved in the scenario, the framework observes them. Then, it applies corrective actions to promote patterns of behaviour that are compliant with the regulations and protocols to be followed. That is, in exceptional situations where actors consider it is beneficial for the society, actors have the option to violate regulations and protocols.
- Expressiveness: Our framework supports the definition of regulative norms via obligations, permissions and prohibitions. It supports sanctions (either positive or negative) and repair actions to promote compliant behaviour and deal with undesirable situations derived from non-compliant behaviour respectively. It supports constitutive norms (providing institutional interpretations of brute facts) and institutional powers, in the form of constitutive powers and constitutive powers.
- Dynamism: The set of regulations and protocols governing the system can be updated. Updates can be performed at run-time, while the system is checking if regulations and protocols are abided, and inferring a normative state consistent with the updates performed (*e.g.,* removing an obligation can remove the sanctions associated to previous violations of the obligation).

This *Chapter* is structured as follows. First, in §5.1 we introduce the *AVICENA* project and the high level motivations of the research being done on Ambient Intelligence for supporting independent living. Then we present proposals existing in the state-of-the-art that have properties common to the COAALAS (§5.2) approach we use as basis. Please notice this analysis focuses on works grounded on Ambient Intelligence for supporting independent living. For an analysis on Normative Systems applied to health-care see §2.4. Then in (§5.3), COAALAS is used to model *AVICENA*'s social structure, effectively providing social-awareness to *AVICENA*'s components. Later, in §5.4 we introduce examples of the norms defined to support the scenario presented, effectively demonstrating the application of NoMoDEI and how it can be used to extend COAALAS and support the *AVICENA* approach. Later, in §5.5, we document the implementation of our test framework focusing on the architecture and the norms resulting from the tests in our scenario. Finally, in §5.6, we compare our approach with other Normative Systems applied to health-care and draw some conclusions.

## 5.1 THE *AVICENA* PROJECT

In the scope of this *Chapter* we focus on scenarios where elder users with chronic conditions, physically or cognitively impaired, have to comply with the treatment prescribed by a doctor. Such scenarios can get especially complex due to a high and uncountable number of potentially probable circumstances, *e.g.*, the combination of several treatments that impose a temporal order on the doses, lack of user's discipline on taking the medicines during the correct interval, delays on the delivery of the medicines, lack of communication between the user and the doctor, non-compliance with dietary or activity habits, and so on.

In such scenarios, the primary goal of our approach is to provide enough support to enable a change in the users' (including elders, doctors, health professionals among other stakeholders) non-compliant behaviors by actively engaging them in the drug intake and treatment follow-up task. With this purpose in mind we introduce the design and proposed implementation of a social-norm aware m-Health solution for treatment adherence that includes a smart pill dispenser. *AVICENA* our m-Health solution, based on the concepts proposed in COAALAS [GSGGÁN11], will support the elderly or people with chronic conditions to manage their daily doses of medication and treatment indications while presenting the following three properties:

- *Social awareness*: The device is connected with other assistive devices and with relevant actors (such as doctors, pharmacists, caretakers and other health professionals, relatives, *etc*) for helping the patient following his medical treatment.
- *Autonomy*: The system can react to changes in the physical or social environment without requiring human intervention. Furthermore, it should be able to react to simple changes in the scenario autonomously (*e.g.*, a change in the scenario implies the pill dispenser is not filled by the patient anymore, but by a care giver).
- *Normative awareness*: The system performs its task while following a set of specified behavioural patterns. However, due to its autonomy, the m-Health solution has the option of breaking the patterns, provided it considers it will be in the benefit of the society (*e.g.*, if an incoming stock break is detected).

### 5.1.1 The COAALAS project

The COAALAS project aims to create a society of organisational aware devices (typically sensors and actuators) that are able to adapt to a wide range of AAL situations. This approach is in-line with the IoT initiative [VF13]. COAALAS models the device network around the user as a society, including the set of behavioural patterns the devices are expected to follow. COAALAS effectively supports smart assistive tools that integrate human actors with the surrounding devices, contributing to the state-of-the-art in semi-autonomous and intelligent devices for elder people by allowing the devices to be both social- and norm-aware.

The mid-term objective of COAALAS was to integrate a wide range of sensors and actuators in a domotic setting, in order to transparently assist the user in their daily activities, while keeping all the participants of the healthcare workflow involved. The first design and implementation of such a sensor/actuator is the social electronic reminder for pills [GSGGAN$^+$12], which tackles the supply of the required stock of medicines to a user with difficulties to leave their house, while supervising that he follows the medical treatment prescribed by his doctor, not missing any dose due to forgetfulness or taking it at the wrong time due to confusion.

### 5.1.2 *AVICENA*

*AVICENA* proposes the development of an innovative m-Health platform and well-tailored personalized services to substantially improve chronic patients' medication and treatment adherence. *AVICENA* platform incorporates:

- **a Smart pill dispenser** that provides the medication at the prescribed times inspired in *SPiDer* [Mor13], controls missed doses with the included sensors, controls the drug stock and contains a reasoning engine offering Smart services. A prototype version of the pill dispenser can be seen in figure 5.7,
- ***AVICENA* mobile app** empowering users with the ability to self manage their treatment obtaining tailored information and feedback depending on their treatment concordance,
- **a new care model** involving all the stakeholders in the chronic treatment process and in the assessment and management of the treatment adherence,
- ***AVICENA* social network** that will connect all the stakeholders in the care process like patients, clinicians, caregivers and pharmacists.

In figure 5.1 we depict the *AVICENA* architecture. *AVICENA* offers the users the opportunity to self-manage their personalized therapeutic programme under medical supervision at home using the *AVICENA*platform. This allows patients to achieve their treatment goal(s) in their preferred environment, improving motivation, and providing objective assessment of the personalized protocol and its outcomes during all the process. Furthermore, it provides a unique opportunity to gather a large amount of data on adherence to therapeutic regimen outcomes for extended periods of time, that can be shared to increase knowledge on users' lifestyle and evolution, to predict potential risks or benefits and to avoid the cold start effect when new users start to use the *AVICENA* platform. *AVICENA* will be able to provide timely and precise indicators of patient adherence and produce relevant reports that will help clinicians to improve the personalization of the treatment and react when necessary. These reports will be produced from the individual user's interaction with the *AVICENA* platform and processed partially in situ. Health outcomes cannot be accurately assessed if only resource utilization indicators and efficacy of interventions measure them. The population health outcomes predicted by treatment efficacy data cannot be achieved unless adherence rates are used to inform planning and project evaluation.

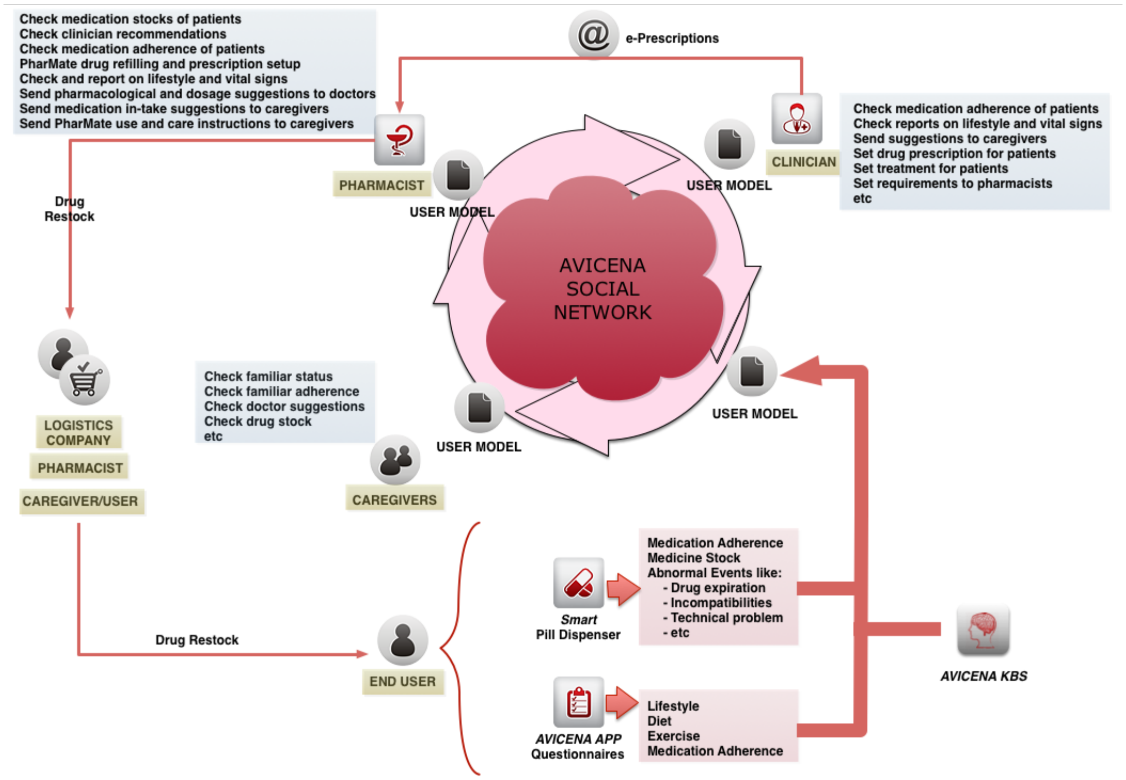



Figure 5.1: *AVICENA* General Architecture

The data generated by the use of *AVICENA* platform would help to overcome this situation, providing timely precise indicators of patient adherence that will help clinicians to improve the treatment personalization.

To improve adherence and develop target interventions, it is important to address the specific reasons why a patient is not able or willing to execute the treatment plan. From this perspective, interventions should be personalized or tailored to address individual needs and beliefs. The definition of tailoring describes the features that make tailored health messages different from other approaches: It is assessment-based and as a result the message can be individual-focused. In other words, tailoring is based on gathering and assessing personal data related to health outcomes or several determinants in order to determine the most effective strategy to meet that person's needs. With these characteristics, a tailored message is able to provide personal feedback, commands greater attention, is processed more deeply, and is perceived as more likable by patients than a general message. Because of these possibilities, tailored health messages are also more likely than generic information to be read, remembered, and viewed as personally relevant. Technologies can be used to tailor health messages to the personal situation of the patient and might therefore contribute significantly to the development of tailored message strategies. The advantages of tailored message strategies can contribute to the incorporation of interactive and continued self-monitoring, feedback, and information exchange, which play an increasingly important role in changing patients' behaviour. *AVICENA* offers the opportunity to solve the patient's non-adherence by encouraging self-management of the treatment and making possible the continuity of therapeutic regimen; it focuses on developing innovative control mechanisms for collaborative, adaptive, dynamic and user centred medical concordance assessment and management systems at preferred environments and highly cooperative, intuitive patient/machine/pharmacist/doctor interfaces over a network.

The *AVICENA* approach and its tools, as they are envisaged, belong to a new generation of intelligent services that are designed to be safely deployed and used by patients alongside other assistive devices to support their owners in their daily life and improve their concordance, adherence and persistence to a therapy. Consequently, this will augment their wellbeing, support adherence and interaction with their caregivers, pharmacist and doctor. In this sense, the *AVICENA* platform will be a powerful extender of user's capabilities and serve society by reducing care costs and providing valuable knowledge about the everyday people's experiences in dealing with a therapeutic regime. For effective provision of care for chronic conditions, it is necessary that the patient, the family and the community who support him or her play an active role [HO03]. Social support, *i.e.* informal or formal support received by patients from other members of their community, has been consistently reported as an important factor affecting health outcomes and behaviours. There is substantial evidence that peer support among patients can improve adherence to therapy while reducing the amount of time devoted by the health professionals to the care of chronic conditions. For this reason, the *AVICENA*platform backbone is envisaged as a *social network* that connects all the community related to the healthcare ecosystem: patients, caregivers, clinicians and pharmacists. *AVICENA* deploys an integrated self-managed adherence support platform, which will be flexibly adjustable to various requirements and needs that are associated not only to the user treatment and socio-environmental characteristics, but also to the various particularities of the different national healthcare system. The EU member states healthcare systems are very heterogeneous, different services are offered and in a different way. For instance some countries (*e.g.* Sweden) or only some regions inside countries (*e.g.* Catalonia in Spain) have integrated the *e*-Prescription to the public healthcare system while others (*e.g.* Italy) have not. This means that *AVICENA* has to cope with a variable context that will require adaptation and tailoring. That is why the *AVICENA*solution is built in a modular way, so the different modules can be stacked depending on the need of the context. In figure 5.2 can see the *AVICENA*'s layered and modular structure:

- At the top we find *AVICENA* users, whose participation depends on the context setup. The basically required are a healthcare representative and off course the patient.
- The *AVICENA* Social Network that enacts a role of communication channel among all the stakeholders. The level of connections depends on each case: doctors may be connected to their patients, pharmacists may be connected to patients and to the related doctors, caregivers may be connected to the patients and patients may be connected to all the rest, even other patients. In all cases, legal and ethical regulations will be enforced to ensure privacy and safety of health sensitive data.
- The Intelligent Layer is a Knowledge Based System that will gather information from all the rest of the modules and devices in order to process it, extract relevant knowl-

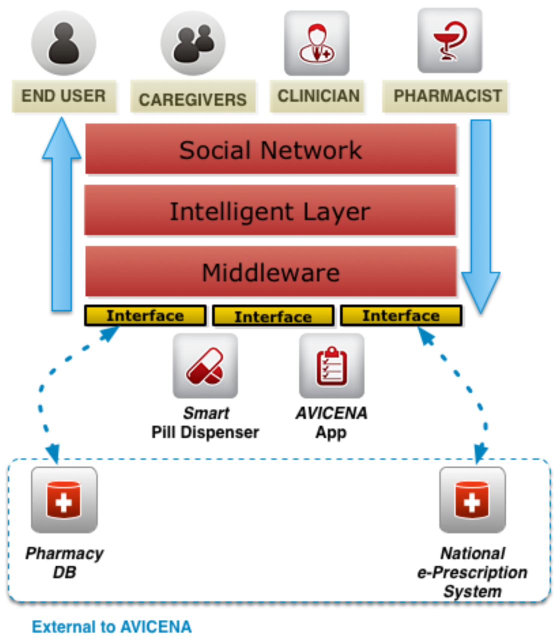



Figure 5.2: *AVICENA* Layered Architecture

edge and proactively enact services for the *AVICENA* platform. These services may be among others the generation of health summary reports for the clinicians based on exploitation of data generated by the system, warnings about drug incompatibility, detection of (un)healthy patterns in the patient, generation of motivational feedbacks to the user to stay adherent to treatment, patient pharmacological history, *etc*. The generated information could also, in further developments, be included in the patient EHR in a context of interoperability.

- The Middleware Layer will interface the different systems and devices that *AVICENA* could potentially integrate, normalizing the information in order to pass it to the platform. The Smart Pill Dispenser or the *AVICENA* app are two of these systems, but it also could integrate national systems like e-Prescription systems, EHRs, pharmacy databases, *etc*.
- The Smart Pill Dispenser is the novel device that will be in charge of the pharmacological aspects of the *AVICENA* project (storage, dispensing, stock control, *etc*). This device will be integrated in the *AVICENA* platform, taking advantage of sensor information and intelligent services.
- The *AVICENA* app is a mobile application that will enable the patient to self-manage all the different aspects of his chronic condition. The application will gather information from the patient regarding lifestyle, dietary habits or other treatment aspects through the use of questionnaire-like quizzes. The application, integrated with dispenser intelligent services and the *AVICENA* social network will offer the patient the opportunity to connect with other users in the same condition, compare experiences

and to some extent compete in healthy habits and adherence ratios. The app will also provide with training information and tips in order to engage the user in healthier behaviours and to be concordant with his treatment.

One of the main concerns when developing assistive technology for the elderly is the need for respectful interaction with the end-user and this will be contemplated at all times in *AVICENA* development. Inputs from the end-users will be captured through a range of methods like interviews, focus groups and recorded observation of usability performance. Inputs coming from other relevant stakeholders (caregivers, clinicians) will also be considered.

### 5.1.3 A sample scenario

To illustrate the aims of *AVICENA* we present a representative usage scenario:

Carlos, 63 years old, suffered three months ago from onset of Congestive Heart Failure. He has been discharged from the hospital after his dyspnoea and he goes back home where he lives alone. His health situation requires a long-term specialised treatment of combinations of a number of medicines to be taken three times a day, a light salt free diet and some exercise. Carlos is not really good at remembering when to take medications and overlooks to take some of his medication doses, which could affect his recovery and his delicate health status. He is concerned about this issue and he agrees under the doctor guidance to join our program. Therefore, he subscribes the obligations derived from norm $N_3$ (see *Norm* 5.10).

Carlos undergoes a complete assessment (physical, cognitive, emotional, and functional) - performed by his medical staff - in order to obtain his clinical profile before entering the *AVICENA*assessment and management program. When the clinical profile is defined, the system - which contains a library of lifestyle recommendations, exercise routines and diets, previously created by an interdisciplinary team of specialists and classified by disease (or combination of them) -proposes an individual treatment program, specifically created by combining appropriate components of the different programs in the library, on the basis of Carlos' clinical profile. The clinician in charge controls the proposed program and - after minor changes - assigns the user to the program. The selected program consists on a medication treatment and series of lifestyle recommendations. The doctor prescribes the medications and generates the corresponding electronic recipes. On Monday morning Carlos stays at home and receives the visit of a technical worker of the *AVICENA* project that deploys the smart pill dispenser (SPD) and the mobile app, explaining to Carlos *how* to use them and secure his agreement to follow the expected daily routine for using these devices to comply with the treatment monitoring. The *AVICENA* technician creates a profile in the *AVICENA* Social Network for Carlos (with access to the services run at SPD interface). Carlos profile is linked to his doctor profile, to the closest pharmacy enrolled in the healthcare consortium and to the profiles of Carlos' relatives of choice. The *AVICENA* app is also connected to the *AVICENA* Social Network representing Carlos, generating information feeds with regards his treatment compliance. The system is set up.

John is a pharmacist that conducts his business one street away from Carlos home. He is enrolled in the healthcare consortium and daily controls using the *AVICENA* Social Network the medicine supply of the users he has linked in his profile. Today he sees that a new user has joined his network, Carlos, and John has received a first posting coming from Carlos' profile triggered by his SPD asking for a full medicine resupply. John checks the electronic prescriptions identified through Carlos's ID and prepares the package to be delivered with the medicines. The logistic company, specially trained to collaborate on the healthcare consortium, receives the package and brings it to Carlos' house and then proceeds to refill the SPD. In the restocking process, the SPD stores the expiration date of the medicines, its composition (to check possible allergies or incompatibilities) and extracts the dosage routine from Carlos' profile that was previously introduced by his doctor. Every day the SPD reminds Carlos to take his medication and provides him the prescribed amount in the tray, the *AVICENA* app also provides reminders. The pills are delivered one by one, providing specific instructions depending on the kind of medication. Using its embedded sensors, when the SPD detects that one pill has been taken, provides the next until all the medications of the treatment have been provided. If Carlos does not take the medication for any reason, the SPD retires the dose from the tray and the event is logged, knowing specifically which medications were missed at that time. Also a new entry is generated in the *AVICENA* Social Network profile, notifying about the missing dose, so the caregivers receive the update when they login in the *AVICENA* social network (or via push notification in the mobile app). The *AVICENA* mobile app reminds Carlos to perform the periodic health check, so he uses the integrated sensors to record his health vital signs, and proceeds to answer the questionnaires that are related to the treatment program recommended by his doctor (see *Norm* 5.3). The questionnaires are friendly and easy to answer to avoid dullness and frustration. The *AVICENA* mobile app provides a range of user-friendly interactive screens that provides Carlos with direct feedback on his progress (*e.g.* motivational messages, information about his treatment, educational feedback about his chronic condition) and allows him to communicate directly through voice, image and video to his clinicians and carers.

Olivia is the doctor that Carlos has assigned and who is following his treatment. Once a week, she connects to the *AVICENA*'s Social Network and checks the evolution of all her patients (see *Norm* 5.3). She can connect to Carlos's vital status at any time to review the answers that he provided to the regular questionnaires about his lifestyle, diet and exercise and also check *how* the medication adherence has been going, detecting missed doses or other unexpected events. She can adjust the questions she wants Carlos to respond to and can create and upload new questions and measures if these are necessary to manage Carlos's health status. She can also be alerted directly to her mobile phone, email or through the web interface if any important breach or adverse effect is identified. On this basis, she can reconsider the prescribed medication observing the health evolution and the other observed parameters; she can send recommendation messages to the caregivers and directly to Carlos on the mobile app, if needed also. In this particular case, Olivia decides to reduce the dose of a specific drug as Carlos health is performing really well. She makes a new electronic prescription and introduces the changes in Carlos profile. Meanwhile in Carlos' home, the SPD *detects* and acknowledges the change in Carlos' treatment and posts a message in John's profile (the pharmacist) asking for a drug substitution. All actors' capabilities are depicted in Figure 5.6.

In the scenario presented in §5.1.3 the following elements were defined:

- the set of doctors as $[D_1, \dots D_k] \in \mathcal{D}$
- the set of patients as $[A_1, \dots A_i] \in \mathcal{A}$
- the set of pharmacists as $[\phi_1, \dots \phi_i] \in \oplus$
- the set of medications as $[M_1, \dots M_k] \in \mathcal{M}$
- the set of prescriptions as $[R_1, \dots R_k] \in \mathcal{R}$

- the set of medical treatments as $[T_1, \dots T_k] \in \mathcal{T}$
- the set of times as $[\tau_1, \dots \tau_k] \in \mathrm{T}$
- the set of substances as $[S_1, \dots S_k] \in \mathcal{S}$
- the set of medical records as $[\rho_1, \dots \rho_k] \in \mathrm{R}$
- the set of questionnaires as $[Q_1, \dots Q_k] \in \mathcal{Q}$
- the set of devices as $[\delta_1, \dots \delta_k] \in \mathrm{D}$
- a competent authority $C$ representing the institution responsible of the system

Please notice, for simplicity we are using sub-indexes only when they are essential. For instance, if an example contains two patients, we will refer to them as $Pt_i$ and $Pt_j$. However, if only one patient is in the example, we will refer to him as $Pt$. Following this idea, we are also using super-indexes only when they are essential. For instance, if it is irrelevant which city the doctor is assigned to, we will refer to him just as $Dr$.

### 5.1.4 Research questions

Summarising, four research questions addressed in this *Chapter*are:

1. Can a social-norm aware m-Health system help elders adhere to their medication prescription? (*i.e.* daily take all the doses)
2. Can a social-norm aware m-Health system help elders adhere to their medication regime? (*i.e.* daily take all the doses at the prescribed time and with the correct order)
3. Can a social-norm aware m-Health system help elders adhere to their medical treatment in a broader spectrum? (*i.e.* diet habits, lifestyle, exercise)
4. Can a social-norm aware m-Health system help the other users involved in the treatment workflow take care of unexpected events?

## 5.2 STATE OF THE ART

This section presents a state of the art on monitoring and assisted living devices for patient and elder care. The section will analyse device capability with special focus on comparing them to proposed and existing social and norm aware devices. For a state of the art on electronic institutions applied to e-health, please see §2.4. For an analysis of these systems compared to our proposal, please see §6.3.2.

Ambient intelligence [ACRV13] [Sad11] (AmI) is the vision of a future in which environments support people inhabiting them. Different kind of AmI applications have been developed, including continuous monitoring, assisted living and others. The market for health monitoring devices is currently characterised by application-specific solutions that are mutually non-interoperable and are made up of diverse architectures [VF13]. This section presents a short survey on the existing work in the area of Ambient Intelligence for supporting independent living, with special emphasis on the works focused on facilitating activities of daily living (ADL). We specially focus in those related with the intake of a prescribed medication. Special attention is given to COAALAS that has been selected as basis for the work presented in this document.

### 5.2.1 Medication prescription and regimentation

AT can be effectively used for guiding elders with their prescribed treatments, avoiding major problems such as non-compliance with the treatment and adverse drug reaction.

Several devices are available for helping patients manage their daily doses of medication. They range from simple pill containers with multiple compartments that can hold a month's supply to intelligent pill dispensers [Inn12] with an alarm function which can detect when the patient takes the pill, and that can be telematicaly programmed in case the treatment changes. However, these kinds of devices tend to have a static encoding of their functions, and are unable to react to changes in the environment (*e.g.*, they will keep on dispensing the pills even if the patient is on holidays away from home) and autonomously react to potentially dangerous situations (*e.g.*, the dispenser is about to run out of supply for a given pill). Furthermore, to the best of our knowledge none of these devices takes in consideration the important role that third parties may have in the activity. For instance, the prescribing doctor scheduling a visit with the patient when the treatment finishes, a delivery company refilling the dispenser when it is about to run out of medication, or the patient's personal computer displaying reminders when it is time to take a given medication. Nor they reflect the social constraints that apply in the relation between the user and the other actors. For instance, forbidding the delivery company employee from entering the user's home if the doctor considers the user capable of autonomously refilling the dispenser.

### 5.2.2 Agent-based healthcare systems

In [AN04] ECA (Event-Condition-Action) rules are used for Smart Homes that support assisted living for the elderly. A basic interpretation of the ECA rules is that, on detecting certain events, if certain pre-conditions are satisfied, then a given set of actions are to be enacted. By using rule-based systems and other Artificial Intelligence (AI from now on) techniques, devices and hardware-oriented technologies for Smart Homes can be augmented and enriched. With that goal in mind, authors propose connecting the devices to a central monitoring facility that performs all the reasoning. This approach differs from the rest in the sense that devices show a complete lack of intelligence, leaving all the reasoning to a central component, effectively preventing coordination and cooperation among the agents representing the different devices.

A similar work [SS09] proposes using abductive logic programs for the reasoning process. Abductive logic programs provide active behaviour, just like the ECA rules, but they also provide added declarative semantics and a extensive background knowledge available via the logic programming. For instance, this approach allows for easily applying preferences to the reminders issued to the user. Both works present a higher system adaptability, allowing even for a customization that adapts the system to the preferences of the user. However they lack the coordination among different agents that would allow the system to autonomously recover from a failure if one of the agents stops working.

*Robocare*

Robocare [COS02] is a project deployed on a domestic test-bed environment that combines a tracking component for people and robots and a task execution-supervision-monitoring component. The system is composed of several software and hardware agents, each providing a set of services, and an event manager that processes requests to the different services and directs them to the appropriate agents. The system also includes a monitoring agent, with knowledge of the assisted person's usual schedule. In order to coordinate all the agents and monitor user's behaviour heavy computational processes take place, limiting the tested scenarios to non-crowded environments, where only 2-3 persons and only a small portion of the domestic environment are monitored. What is more, the expected schedule is non dynamic and small justified deviations (*e.g.*, relatives visiting the user) are currently detected and corrected.

*Independent LifeStyle Assistant*

The AHRI (Aware Home Research Initiative) [KPJ+08] is a residential laboratory for interdisciplinary research where several projects have been evaluated. The most relevant one is the ISLA (Independent LifeStyle Assistant) project [HKM+04], that passively monitors the behaviours of the inhabitants of the residential laboratory, alerting relatives in case of potentially dangerous situations (*e.g.*, the user falls). The ISLA project presents two main innovations with regards to the Robocare project:

- Agents autonomously interact within them in order to achieve their goals, without the need of an event manager agent that coordinates them. However, in order to transform context-free perceptions provided by the agents into context-aware perceptions, a centralized coordinating agent is used.
- Agents are able to learn schedules based on the daily tasks performed by the inhabitants. Models are built, reflecting which devices are triggered as a result of the performance of which activities, and alerts are raised whenever an unlikely activity takes place. Therefore, instead of using generic static schedules for the users, the schedules are built dynamically based on user's detected behaviour. However, once a schedule has been learned, the user is not able to deviate from it without raising an alarm.

Evaluation of the ISLA project presents two main conclusions:

- The need for coordination of the agents and centralized control outweights the benefits of the distribution and independence of components agents architectures provide.
- Partial observability of actions performed by the inhabitants is a problem, specially when plans are abandoned due to forgetfulness and reminders need to be issued. Inhabitants do not tend to be in favour of having every of their moves observed.

*MINAmI*

In the scope of the MINAmI project[NFKG07] a qualitative study of three ambient intelligence scenarios is reported, being the most relevant one a scenario that deals with monitoring the taking of medication. In the scenario users are given a smart pillbox, with a cap that counts the number of opening and closing events and a clock. The pillbox can communicate with a mobile phone, that displays the timed record of cap openings and closings. If the users forgets to take his medication for a prolonged period of time, the pillbox sends a notification to a care center. During the evaluation of the scenario, users felt it was too intrusive on their privacy, arguing the data should not be reported to their doctors. They considered relying on such devices for the reminders could weaken people's cognitive abilities, and that such a system would not be suitable for users taking a cocktail of medication rather than just a single medicament, as several pillboxes should be provided. The scenario presented seems to be mainly theoretical, lacking an implementation, and does not provide a fully integration of the pillbox with the rest of the devices in the Smart Home (*e.g.*, the system can notify that the user forgot to take his medication even when the rest of the devices are showing that the user has not been at home on the last 3 weeks, for instance, because he is on holidays).

SPiDer

*SPiDer* focuses in the specific problem elderly face when trying to follow a complex schedule of medications (*i.e.*, the low compliance with complex prescribed medication schedules) and aims to partially solve the aforementioned problems through the use of AT and AI tools [Mor13]. The original system developed by *SPiDer* had three main components: the Smart Pill Dispenser (SPD) which allows to perform the dispense tasks and the sensing of the environment, the Identification Access Management (IAM) which allows to monitor and manage the entrance and exit of the house, and the MAS which integrates both the SPD and the IAM components and adds new features as communicative tools between the system and the outside world. There it was a physical implementation of the SPD (see figure 5.3.


Figure 5.3: The Smart Pill Dispenser of the *SPiDer* project

*HomeRuleML*

Decision support rules can be applied to smart environments. They will be able to detect when the patient is performing a task, how he is performing it and at which stage the task is. Research on smart home environments and Ambient Assisted Living has often lead to the development of expert systems and rule-based systems for the analysis of the information recorded within the smart environment. However, according to the authors of HomeRuleML [HND+07], few attempts have been made to achieve a cross-system standard for decision support systems in smart environments (where both environmental and sensor data are taken into account), effectively allowing for storing and exchanging decision support rules in the context of smart environments. HomeRuleML proposes a standard for a wide and freely accessible set of rules, in the scope of smart home environments, which can be openly exchanged within the research domain and beyond.

The main idea is providing a model of rules for smart home environments and Ambient Assisted Living with the following properties:

- Simple to design: Because often the set of people with the technical knowledge (capable of designing conceptually complex rules) does not necessarily intersect with the set of people with the medical knowledge (capable of understanding the needs of the patient). The idea is avoiding the scenario where rules are created by developers and not by the people with the actual insight and domain knowledge.
- Expressive: So they have sufficient details for the technical developers to adapt and configure the rules to the available resources (*e.g.,* sensors) in the environment. Not too high level so as to lack detailed information about sensors in the environment and the environment itself. In other words, provide enough information so as to bind sensor data to a specific sensor.
- Flexible: Possible to integrate the rules dynamically in the smart environment.

  HomeRuleML proposes an interesting standard for exchanging device and event information in the scope of smart home and Ambient Assisted Living environments. However, the proposal presents two main drawbacks compared to COAALAS. First, the proposal is completely device centric, not presenting alternatives for modelling other actors involved in the scenario and not related to devices (*e.g.,* patient, doctor and other healthcare professionals, relatives, etc.). Therefore, it lacks the social layer COAALAS provides. Second the proposal is centred on one modelling layer, the lower one to be precise, as they aim to provide enough information so as to bind sensor data to a specific sensor. Lacking abstraction layers above HomeRuleML providing abstract sensor definitions make it difficult to automatically adapt the system to changes in the environment or in the sensors (*e.g.,* new sensors are purchased, the patient changes location, some sensors stop working temporally or definitively, *etc.*) and therefore, the approach does not seem suitable for dynamic scenarios.

*Planning in robot ecologies*

Robot ecologies [SBG+08] are a growing paradigm in which several robotic systems are integrated into a smart environment. Such systems hold great promises for elderly assistance [Jap]. However, the issue of planning and coordinating the activities of these systems must be tackled. This is not trivial as it requires taking into account temporal and information dependencies among different parts of the ecology, exogenous actions and multiple, dynamic and sometimes conflicting goals. The work in [DRSG+14] describes a planner designed to cope with such challenges that is able to:

- Cope with different, concurrent and even conflicting goals that can be added at any time.

- Perform high-level reasoning based on the different goals to be achieved. Map the output of the reasoning process into low-level actions to fulfil the system requirement's.
- Coordination of different robotic units to perform heterogeneous actions with support from the environment.
- Usage of common resources, time synchronisation and causal and information dependencies.
- Able to rearrange activities to maintain the feasibility of a temporal network dealing with unexpected events in a very high-level way.

The planner's capability is depicted in a sample scenario where the robot *Coro* is working on an elder residence. Laundry for patient *Sven* has just finished, and the integrated system sends a message to *Coro*, so the robot can pick up the clean clothes and deliver them to the patient's room. The robot interacts with the environment (electronic doors, elevator, *etc.*) in order to pick-up the clean clothes. However, on his way to *Sven's* room, *Coro* receives a message from the kitchen. The meal for patient *Gunilla* has been prepared, and must be delivered to the patient's room as she is in bed recovering from an operation. The planner is able to identify the goal to deliver the meal being more important than the one for delivering the clothes (as a cold meal would not fulfil patient's requirements and the clothes can wait), so it generates a new plan for *Coro*. The robot has to leave the clothes in a safe place, pick up the meal, deliver it to *Gunilla* and only then, go back to the plan for delivering the clothes to *Sven*.

The planner proposed shows some basic capabilities for dealing with dynamic goals in environments where unexpected events can occur. However, it does not provide flexibility at the level COAALAS does. The planner is not able to define roles, norms and dependencies between the different stakeholders in the scenario. Therefore, roles can not be reassigned (the planner has no means to schedule the delivery of the clothes or the food if *Coro* stops working) as there is no social structure to support such changes. Norms are absent, the notion in *Coro* that delivering food is more important than delivering clothes is somehow hard-coded, versus the explicit norm *It is forbidden to deliver cold food* COAALAS can provide. The proposal is a centralized planner that can make the system fail if one component (*i.e.* the planner) fails, whereas COAALAS proposes a high-level model every component can understand. Finally, there is no high-level definition of the system non-IT experts can adapt to changing situations (such as, more robots being available).

*A model driven engineering process of platform neutral agents for ambient intelligence devices*

The work presented in [AAF14] is based on one of COAALAS premises: *Ambient Intelligence devices should be enhanced with software able to react and adapt to user actions or events that have occurred in the environment*. To achieve this, they rely on model-driven engineering techniques, an approach for software development that promotes the use of models to formally represent domain-specific concepts. The idea is obtaining the final software through the transformation of different meta-models defined at different software abstraction layers. Model-driven engineering approaches allow to specify high-level concepts in a platform independent model and transform it to a particular software implementation model, effectively bridging the gap between design and implementation.

The idea behind this work, is transforming the high level model to agents based on the *Malaca* agent platform [AF09]. On the one hand, this differs from the approach taken in COAALAS in the sense we provide means for agents to understand the high level models, rather than relying on transformations that automatically generate agent code. This allows for agents (and by extension the different Ambient Intelligence devices operated by them) to be organizational-aware effectively understanding and reasoning about their position in the organization (that is, the high level model). On the other hand, the approach does not present a high level model as expressive as the COAALAS architecture, as it lacks core concepts such as norms, roles and a clear social structure.

*THOMAS and GerAmi*

THOMAS [BFPLC10] provides a set of modular services extending the FIPA standard [BPR99] to deal with organizations. This is done via a central component known as *Organization Management System* (OMS). Roles, organizational units and norms can be effectively created via the *OMS*.

The THOMAS architecture has been applied to several projects on ambient assisted living and patient-care, and GerAmi [CBA08] is an example of such projects. In GerAmi a system with a geriatric agent to optimise work schedules and provide up-to-date patient data is presented. The idea is combining case based reasoning with planning capabilities to achieve such goals. The main focus is to include agents' into persons to enhance communication and work scheduling, effectively making nurses' working hours more productive. This will effectively give nurses more time which they can use to caring for special patients, exercising patients, helping them with their leisure activities, talking to patients or their families, or even enhancing nurses' knowledge. The system includes five roles: Security, nurse, doctor and patient. The system, does not include both human and computational agents, because agents with human names are just sensors monitoring the person. For instance, patient agent validates patients location, monitors state and sends a copy of its memory to perform back-ups of patient's data. Nurse agent manages nurses' working days, managing profiles, tasks, available time and resources. The system provides both robustness and security from a computational perspective (*e.g.,* hourly back-ups and a raid system with a robust WLAN). However, the system does not define potentially dangerous situations and how to react to them. It also does not reorganise in case society changes (*e.g.,* relative acting as a caregiver or nurse). Even though the THOMAS architecture provides full support for norms, the system does not define a way to define states of the world that should be always achieved or avoided when performing plans, even though such states are mentioned on the scenario presented (*e.g.,* a nurse can not work more than 8 hours).

The *OMS* component in the THOMAS architecture provides a high-level organizational model very similar to the one presented in COAALAS. This allows for creating organizational aware agents that can understand and reason about their position in society. However, none of the articles explaining the THOMAS architecture analysed so far includes a clear example of such organizational definition, and they do not provide a structural definition of the elements. Furthermore, the THOMAS architecture allows for norms can be added and removed, however the effects of modifying such norms on the organization are not addressed, whereas they have been addressed in the architecture supporting COAALAS [GSÁNVSF12].

$COMMODITY_{12}$

$COMMODITY_{12}$ [KBS+13] focuses on providing advice, recommendations and alerts to diabetic patients based on their data, and at the same time assist medical personnel, who is in charge of these patients, facilitating informed and timely decisions. The system consists in two main components: first, a set of devices that collect health-related data (*e.g.*, activity and body signals). Second, a set of personal agents with expert biomedical knowledge that interpret the data via a reasoning process. Then they generate a high level representation of patient's health status that is provided to relevant actors in the scenario (*e.g.*, patients and health care professionals) in the form of feedback reports. The main idea is integrating sensors, intelligent agents, knowledge bases and users within a single system. The work introduces the $\mathcal{LAMA}$ architecture for developing software agents that can reason about a medical domain. Agents are deployed using the GOLEM agent platform [BS08].

$COMMODITY_{12}$ presents an approach to integrate sensors with smart agents (providing an analogous functionality to having smart sensors) in order to interpret sensor information with the biomedical knowledge included in the agents. Unlike other approaches analysed (*e.g.*, THOMAS and COAALAS) $COMMODITY_{12}$ does not define the social structure where agents and devices operate. That is, the $COMMODITY_{12}$ model lacks concepts such as role, objective and norm. Furthermore, even though $COMMODITY_{12}$ agents have expert biomedical knowledge, they are oblivious of the social context they operate in.

### 5.2.3 ALIVE meets SHARE-it: An Agent-Oriented Solution to Model Organisational and Normative Requirements in Assistive Technologies

In a previous work [GSGGB$^+$12] we present an approach to the development of assistive technologies which uses both organisational and normative elements based on the ALIVE [ÁNCPVS09] framework. We illustrate our approach via a scenario where a physically disabled patient (with a particular disability that makes it very difficult for him to leave his house) has to be supplied with his required medication. Our model contains the main actors in the scenario (*i.e.* patient, doctor, health insurance company, pharmaceutic, delivery person, domotic house, intelligent medical dispenser and medical monitor) including both human actors and computational agents socially integrated in the institution depicted in the scenario.

The ALIVE framework eases the design of both the social network built around the patient (*e.g.*, doctors, caregivers, relatives, *etc.* ) and the expected behavioural patterns to be abided by the actors in the scenario. ALIVE is designed as a multi-level architecture, including the three following levels: Organisational, Coordination and Service.

The Organisational level is formalized following the Opera methodology [Dig04]. It contains following concepts:

- Objectives: States of the world pursued by actors. Typically derived from organisational goals.
- Roles: Groups of activity types played by actors (either computational agents or humans). The set of all roles and the dependency relationships between them constitutes the *Social Structure*.
- Landmarks: Represent relevant states of the world regarding the achievement of goals. They are identified by the set of logical propositions that are true on the state of the world represented by the landmark.
- Normative structure: The normative structure imposes patterns of expected behaviour on the set of actors in the scenario.

The coordination level provides the patterns of interaction among actors, effectively transforming the elements in the organisational model into coordination plans known as work-flows. Work-flows will bring the system from the state represented by a landmark to the next landmark, and are formed by chains of tasks. Tasks contain both pre-conditions and post-conditions, defining the state of the world before and after the task is performed. Tasks also contain semantic information that binds them to abstract services in the service level.

The service level provides a pool of services that can be selected for each abstract task composing the work-flows in the coordination level. Service descriptions are provided in terms of OWL-S service profiles [MBH+04].

The main idea of using the ALIVE framework on a health care inspired scenario is providing flexibility. ALIVE's multi-layered architecture allows to automatically trigger changes in the coordination and service levels when new roles, objectives and norms are introduced in the organisational level. Furthermore, the agents in the coordination level provide both organisational and normative awareness to the system, effectively allowing them to analyse and reason about the work-flows before enacting them, discarding the ones that do not comply with organisational norms.

## 5.3 THE *AVICENA*Ś ORGANIZATIONAL MODEL IN COAALAS

COAALAS, as said before, builds on the results of two European funded projects: EU-SHARE-*it* [AC10] and EU-*ALIVE* [ÁNCPVS09]. By combining several state-of-the-art AI techniques (such as Autonomy, Pro activity, Social Behaviour and Adaptability) COAALAS provides a multi-agent platform able to integrate software agents embedded in the AAL devices and human actors. This allows for making AAL devices intelligent enough to organize, reorganize and interact with other actors. The agents embedded in the devices have an *awareness* of their social role in the system – their commitments and responsibilities – and are capable of taking over other roles if there are unexpected events or failures. COAALAS creates a society of physically *organisational-aware* devices able to adapt to a wide range of AAL situations that could have an impact on the user's well-being.

COAALAS builds on top of the *ALIVE* framework, a multi-level architecture – as seen in *Figure* 5.4) it is divided in an organisation level, a coordination level and a service level – that provides support for*live*, open and flexible service-oriented systems. The *ALIVE* framework presents normative structures that allow for easily expressing both expected behavioural patterns and the actions to be taken when the actors involved in the scenario do not comply with these patterns. For achieving this functionality, *ALIVE* relies on substantive norms that define commitments agreed upon actors and are expected to be enforced by authoritative agents, imposing repair actions and sanctions if the system reaches invalid states (*i.e.*, states that are outside of the expected behavioural patterns). Substantive norms allow the system to be flexible, by giving actors (human or computer-controlled) the choice to cause a violation if this decision is beneficial from an individual or collective perspective.

The Organisational level in *ALIVE* contains organizational structures inspired in the *Opera* methodology [DVSD04] by using the following concepts:

- Objective, states of the world pursued by the system and the actors in the system (*i.e.*, daily take the prescribed medication dose).
- Role, Groups of activity types played by the different actors in the system (*e.g.*, patient, caretaker, doctor, *etc*.). The set of roles and the relationships among them form the *Social Structure*.

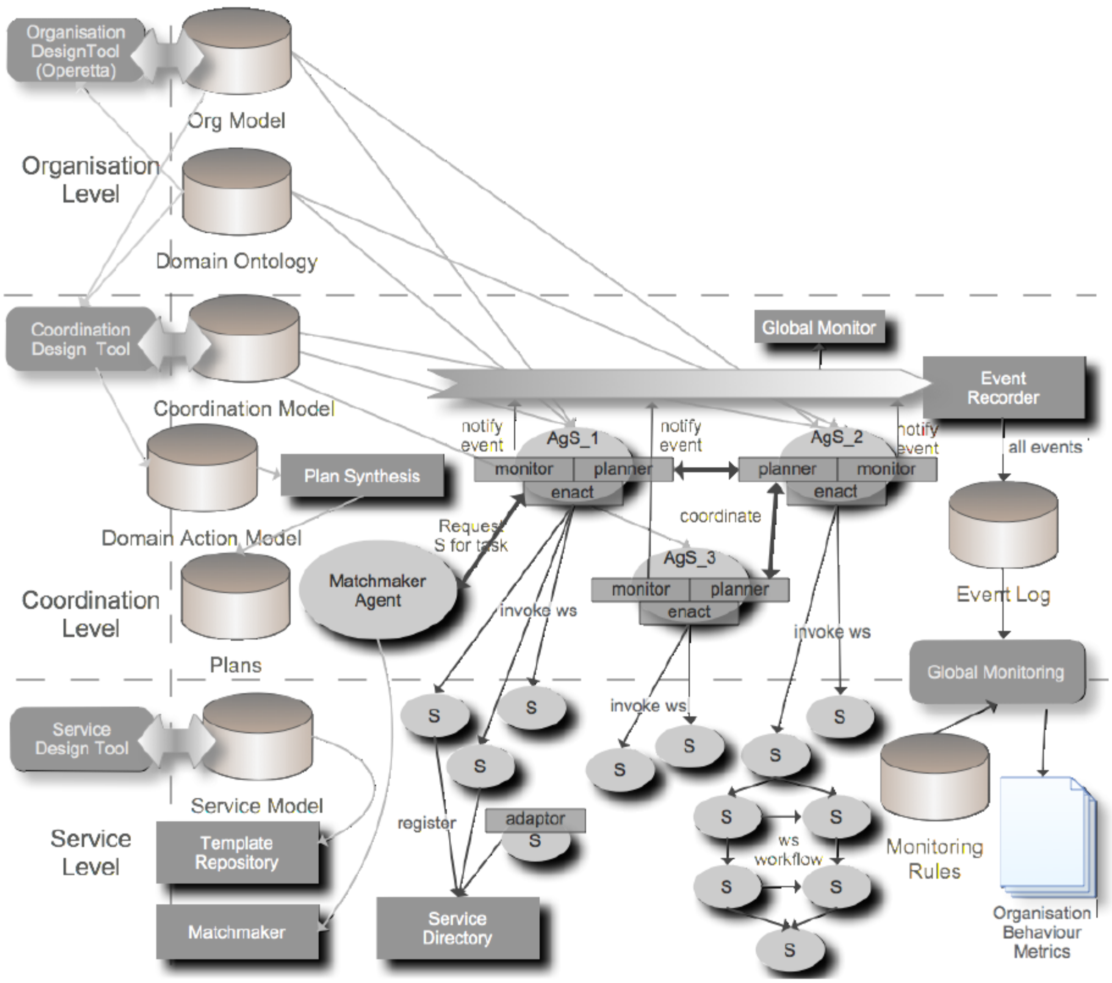
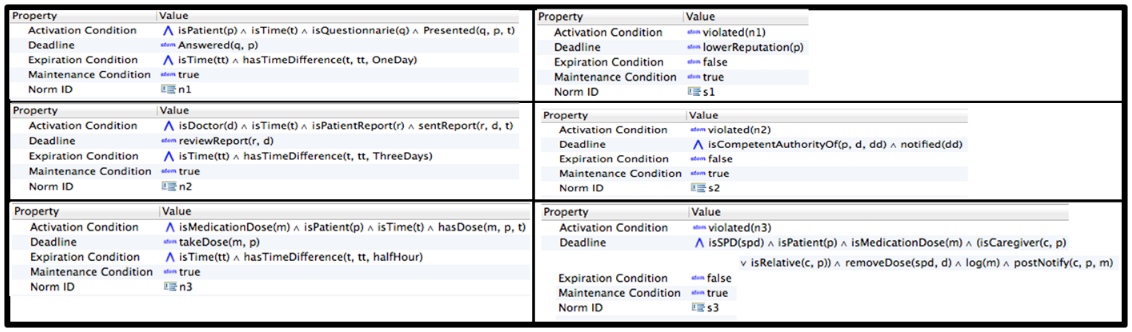



Figure 5.4: ALIVE architecture (S stands for Service)

- Landmark, represent states of the world that are relevant for the achievement of goals (*i.e.*, a medication dose has been provided).

The organisation level supports the definition of substantive norms, adding a normative structure to the social and interaction structures. The norms contain the following main components, expressed using Partial State Descriptions of the world:

- *Activation Condition*: when the world reaches the state specified in this condition, the norm starts to be checked.
- *Expiration Condition*: when the world reaches the state specified in this condition, the norm stops to be checked, and has not been violated.
- *Maintenance Condition*: when the world reaches the negation of the state specified in this condition, the norm stops to be checked, and has been violated.



| Property | Value |
|---|---|
| Activation Condition | ∧ isPatient(p) ∧ isTime(t) ∧ isQuestionnarie(q) ∧ Presented(q, p, t) |
| Deadline | Answered(q, p) |
| Expiration Condition | ∧ isTime(tt) ∧ hasTimeDifference(t, tt, OneDay) |
| Maintenance Condition | true |
| Norm ID | n1 |

| Property | Value |
|---|---|
| Activation Condition | violated(n1) |
| Deadline | lowerReputation(p) |
| Expiration Condition | false |
| Maintenance Condition | true |
| Norm ID | s1 |

| Property | Value |
|---|---|
| Activation Condition | ∧ isDoctor(d) ∧ isTime(t) ∧ isPatientReport(r) ∧ sentReport(r, d, t) |
| Deadline | reviewReport(r, d) |
| Expiration Condition | ∧ isTime(tt) ∧ hasTimeDifference(t, tt, ThreeDays) |
| Maintenance Condition | true |
| Norm ID | n2 |

| Property | Value |
|---|---|
| Activation Condition | violated(n2) |
| Deadline | ∧ isCompetentAuthorityOf(p, d, dd) ∧ notified(dd) |
| Expiration Condition | false |
| Maintenance Condition | true |
| Norm ID | s2 |

| Property | Value |
|---|---|
| Activation Condition | ∧ isMedicationDose(m) ∧ isPatient(p) ∧ isTime(t) ∧ hasDose(m, p, t) |
| Deadline | takeDose(m, p) |
| Expiration Condition | ∧ isTime(tt) ∧ hasTimeDifference(t, tt, halfHour) |
| Maintenance Condition | true |
| Norm ID | n3 |

| Property | Value |
|---|---|
| Activation Condition | violated(n3) |
| Deadline | ∧ isSPD(spd) ∧ isPatient(p) ∧ isMedicationDose(m) ∧ (isCaregiver(c, p) ∨ isRelative(c, p)) ∧ removeDose(spd, d) ∧ log(m) ∧ postNotify(c, p, m) |
| Expiration Condition | false |
| Maintenance Condition | true |
| Norm ID | s3 |

Figure 5.5: Example of norms of the use case defined via the ALIVE editor

- *Deadline*: condition that has to be met before the *Expiration Condition* holds. Typically, used to specify obligations to be met before the norm expires (*e.g.*, obligation to pay taxes before the end of the year).

The increasing dependence on information technologies in health care organisations has increased the interest in security techniques applied to healthcare. Typically, security is concerned with the protection of information from unauthorized access. However, in highly regulated scenarios involving several tasks that must be coordinated by a range of actors, both the roles for some of the actors (*e.g.*, who can act as patient's caregiver) and the way they should or may interact with patients (*e.g.*, who can access patient compliance records) are clearly defined and regulated, and this requires security from a higher level point of view [FD00]. Not only sensitive data must be protected, but the activities carried out by the actors should be controlled, and this implies that the rules and protocols regimenting the medical organisation where the agents operate must be represented, understood by the different agents (either human agents or computational processes) and enforced[Nor90]. Including such rules and protocols into the system will not only enhance security but also social acceptance (from patient's point of view) and professional acceptance (from doctor's point of view).

Via the *ALIVE*'s normative structure system designers can effectively provide a formal description of the norms and protocols regimenting the organization. Such formal description can be understood by the different agents in the system and will allow to:

- Enable the agents operating the MAS to understand the norms and protocols involved in the scenario. This will allow agents to reason about the normative consequences of the actions they perform [PANVS14].
- Provide methods to enforce the norms and protocols, detecting norm violations via a monitoring system [AÁNDVS10] and applying sanctions and repair actions.

*Figure* 5.5 depicts an ALIVE model of the following norms extracted from the scenario presented in §5.1.3. The figure is presented as an example of norms modelled using the ALIVE approach. §5.4 introduces a more exhaustive representation of the norms defined to support the scenario presented in (§5.1.2). The norms are formalized in a metamodel which is inspired in the *ALIVE* framework. Please notice that for a Norm $N_i$ the Sanction for $N_i$ is another norm $S_i$ that is activated when $N_i$ is violated.

**Norm** $N_1$: The patient has the obligation to answer the questionnaires related to the treatment program recommended by his doctor. In the set of norms presented, the patient has one day to fill the questionnaires. This time period can be easily tailored according to scenario requirements. The spirit of this norm is trying to improve system's health monitoring capabilities by encouraging users to fill the questionnaires via a gamification approach [McC12]. The idea is making questionnaires not only user friendly, but also providing a social reward for filling them.

$S_1$: Failing to comply with $N_1$ results in patient's reputation being lowered in the system's social network. This norm can be easily extended (*e.g.*, continuously failing to comply with the norm might result in alert messages being sent to relatives or caregivers).

**Norm** $N_2$: Doctors have the obligation to review an updated patient adherence report. As stated in the set of norms norm presented, the doctor has three days for the review, but again this time period can be easily adjusted to scenario requirements. Updated patient adherence reports are provided to doctors when patient's detected pattern of behaviour (either via questionnaires or via integrated information sources) does not match the expected pattern of behaviour for a prolonged period (typically, non-compliance is detected).

$S_2$: A note is sent to the competent authority. The competent authority is a role controlling the relation patient-doctor. Such role can be fulfilled by another doctor (*e.g.*, the head of department in a hierarchical organization), a caregiver or relative representing the patient or by the patient himself.

**Norm** $N_3$: The patient has the obligation to take his medication. 30 minutes are provided for taking it, but this time can be easily adapted.

$S_3$: The SPD retires the dose. The event is logged including which particular medication dose was missed and at which time. A message is posted on the social network so caregivers and patient relatives are aware of the missing dose.

*ALIVE* provides coordination structures (basically a repository of coordination plans automatically generated from the elements in the Organisational level) that provides actor's patterns of interaction, effectively allowing the system to move between relevant states (*e.g.*, the pill dispenser needs to be refilled, the pill dispenser has been refilled, *etc.*). The coordination structures are formed by tasks containing both pre and post conditions (*i.e.*, the state of the world before and after the task has been executed respectively) and the permissions required for executing the tasks (associated to the different roles in the scenario). A set of organizational aware intelligent agents select a role according to their capabilities and start enacting the plans associated to that role as requested.

Finally, *ALIVE* also includes a service level that maps actions in the environment to abstract tasks. Non-organizational aware agents in the system register their capabilities (*e.g.*, tasks they can perform) via a white pages system and are coordinated by the organizational aware agents to execute the tasks required for enacting the different plans. *Figure* 5.6 provides an example of actor's capabilities in a scenario inspired in an intelligent pill dispenser. A realistic scenario is described in §5.1.3.

### 5.3.1 Related work

Table 5.1 presents a summary of the analysis of the state of the art performed in §5.2 and puts in contrast existing AT w.r.t. COAALAS.

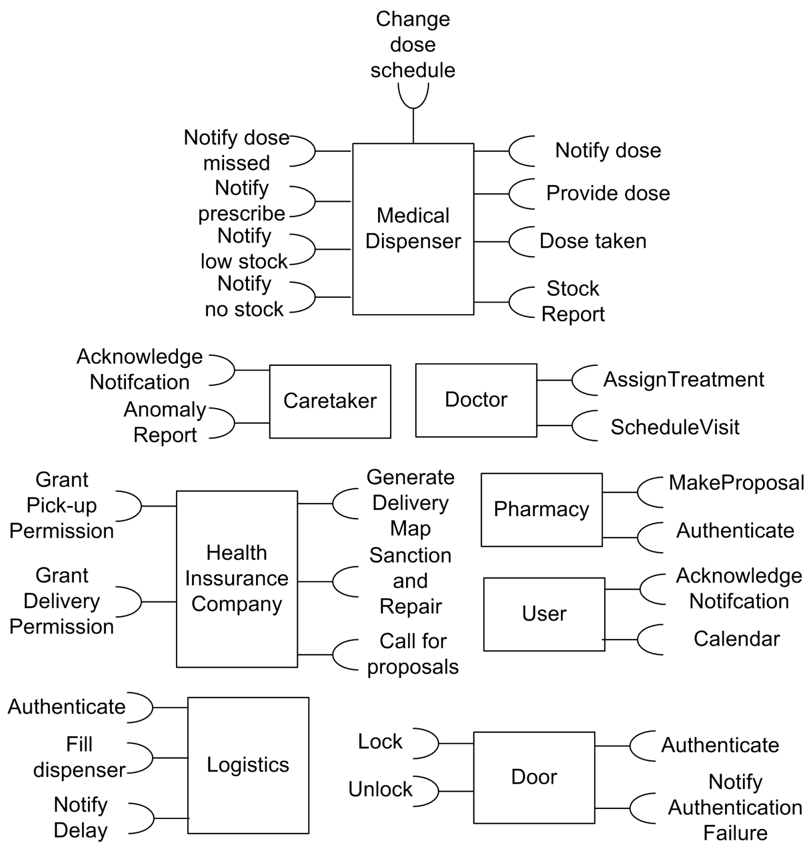



Figure 5.6: Capabilities of the different actors

The uBox is a palm-sized pill dispenser that reminds a patient when it is time to take medication and records when he takes a pill from the dispenser. It is also able to track accesses to the box by other users (typically a health professional refilling the box) via RFID keys. The project is able to ensure patient adheres to his medication prescription. However, it is hard to see *how* it would manage multiple medication doses, ensuring every dose is taken at a given time and in a particular order. Furthermore, the project is not social aware nor able to react to unexpected deviations from patients schedule.

The Robocare project provides coordination between the different actors in the scenario via an event manager, that processes requests for the different services and redirects them to the appropriate agents. As the system also includes a monitoring agent that is aware of the patient's expected schedule, it can be effectively used to ensure the patient adheres to his medication prescription and regime. However, the expected schedule is not dynamic, and only minor deviations can be detected and corrected.

The ILSA project aims at passively monitoring the behaviour of the patient, alerting relatives in case of potentially dangerous situations. Agents in ILSA are able to learn pa-

| Name | Deployment | Social Aware | Prescription | Regime | Adaptive |
|---|---|---|---|---|---|
| **COAALAS** | Smart device integration platform | ✓ | ✓ | ✓ | ✓ |
| uBox | Programmable pill dispenser | × | ✓ | × | × |
| Robocare | Tracking of people and robots | ✓ | ✓ | ✓ | × |
| ILSA | Passive monitoring and schedule learning | ✓ | × | × | × |
| MINAmI | Network-connected pillbox | × | ✓ | × | × |
| Smart Homes | ECA rules | × | ✓ | ✓ | × |
| SPiDer | Smart pill dispenser | ✓ | ✓ | ✓ | × |
| HomeRuleML | Standard for connected devices | × | × | × | × |
| Robot Ecologies | Robots coordinated with smart environments | × | × | × | ✓ |
| Model Driven | Transforming high level concepts to implementation details | × | ✓ | ✓ | ✓ |
| Thomas Architecture | Organizational aware optimization of work schedules | ✓ | × | × | × |
| $COMMODITY_{12}$ | Assistance to diabetic patients with expert biomedical agents | × | ✓ | ✓ | × |

Table 5.1: Comparison of existing solutions

tient's schedule, building a user model as the patient interacts with the different devices. Therefore, instead of using generic static schedules for the patients, the schedules are built dynamically based on detected behaviour, and the resulting schedule is perfectly tailored for every patient. However, once a schedule has been learned, the patient is not able to deviate from it without raising an alarm. The ILSA project is not Social Aware, it can detect some unexpected situations, but is unable to coordinate the different actors in the system for solving them. The project can not be used to ensure patient adheres to a medication prescription and regime, because the behavioural pattern the system learns might not be aligned with them (*e.g.*, the learned expected behaviour is for the user to miss every medication dose).

The MINAmI project presents a smart pillbox, with a cap that counts the number of opening and closing events and a clock. The pillbox can communicate with a mobile phone, that displays the timed record of cap openings and closings. If the users forgets to take his medication for a prolonged period,the pillbox sends a notification to a care centre. The project can be used to ensure patient adheres to a medication prescription, and is


Figure 5.7: The new Smart Pill Dispenser prototype, evolution from SPiDer

able to detect unexpected situations and alert about them. However, it is not able to coordinate the different actors in order to deal with the unexpected situation. MINAmI is not suitable for ensuring patient adheres to a medication regime (specially if a very complex combination of different medications is used) and does not present any social capabilities.

The Smart Homes project proposes connecting the different AT devices to a central reasoning component that is able to issue particular actions when certain events and conditions are met. The project can be effectively used to monitor user's behaviour ensuring he adheres to a medication prescription and regime. However, as all the intelligence in the system is focused on a central component, the project presents no social awareness and its adaptation capabilities are very limited.

*SPiDer* connects a smart pill dispenser and an access management system with a multi-agent system to control and coordinate both components. The project can be effectively used to monitor user's behaviour ensuring he adheres to a medication prescription and regime. However, The *SPiDer* project is not Social Aware in its current status of development. It can detect some unexpected situations, but is unable to coordinate the different actors in the system for solving them.

COAALAS applied to a pill dispenser device provides a social context, help patients adhere to their medication prescription and regime and is able to react to unexpected events at the same time. COAALAS provides a social structure that allows the pill dispenser to coordinate with other devices in order to ensure the patient takes his daily medical dose. For instance, the pill dispenser can coordinate with a device taking care of the patient's schedule to find out if the patient will be away from home when it is time to take his medication. If it is the case (and the medication dose does not require any special conservation procedures) the pill dispenser can dispense a pill before the patient leaves home, and issue a reminder asking the patient to take the pill with him. When it is time to take the medication dose, a reminder will alert the patient, who can take his medication dose even though he is away from the dispenser. In case the patient deviates slightly from the expected schedule (*e.g.*, misses a non vital medication dose), COAALAS will detect the deviation and the pill dispenser can coordinate with other actors (including doctors if required) to re-schedule future medication doses, so the patient complies with his medication regime. Even more, in case the patient deviates heavily from the expected schedule (*e.g.*, misses one important medication dose or several medication doses), COAALAS allows for detecting the deviation and can help the different actors involved in the treatment workflow take care of the situation. COAALAS will not only be able to detect the deviation but will also be able to identify which are the most appropriate actors for taking care of the unexpected event, coordinating them to return the system to an acceptable (*i.e.*, expected) state. The main ingredient of this approach is the consideration of the norm conditions, which define a life cycle of each norm, to infer when a norm is violated. According to [Ald07], the detection of norm violations depends on two properties of inspection to be done:

**Observability:** the conditions or actions can be checked by internal agents, given the time and resources needed;

**Computability:** the conditions of actions can be checked in a feasible and low cost manner.

Still, none of the above mentioned approaches can offer users the opportunity to continue their personalized therapeutic programme under medical supervision at his/her preferred environment.

*AVICENA* offers the opportunity to resolve the patient's non-adherence by encouraging and making possible the continuity of therapeutic regimen, reducing costs to the patient, the caregivers and the health system. It focuses on developing innovative control mechanisms for collaborative, adaptive, dynamic and user centred medical concordance assessment and management systems located at preferred environments and highly cooperative, intuitive interfaces integrating user, machine, pharmacist, and doctor over a network.

## 5.4 NORMS

This section introduces exhaustive examples of the norms defined to support the scenario presented in §5.1.2. The section starts by presenting formal examples of such norms, as defined in ALIVE's normative structure. Then, the section goes on by introducing the effects of the operations defined in §3.4 effectively depicting how norm dynamics will interact with the scenario presented. Please notice that for a regulative norm $N_i$ both the sanction and the repair action for $N_i$ is another regulative norm $S_i$ that is activated when $N_i$ is violated.

### 5.4.1 Norm examples

This section presents formal examples of norms based on the scenario introduced in §5.1.2. Every example consists in an informal explanation of the norm in natural language and a formal model of the norm defined using our formalism.

$A_j$ is going to the pharmacy to pick up some medicines in order to refill his medical dispenser. e-Prescription systems are not available in the area where $A_j$ lives right now, and some of the medicines are dangerous and therefore can only be dispensed with the corresponding medical prescription $R_k$. According to the protocols, the pharmacist $\phi_i$ has the obligation to retrieve the prescription and verify $A_j$'s identity before delivering the medicines. If the pharmacist delivers the medicines without following the protocol, he will be sanctioned with an official warning from the competent authority. The norm is formally introduced in *Figure* 5.8. Please notice how deadline condition sets the obligation to perform a particular action at least once before a particular state of the world is met.

$A_i$ has the obligation to follow the prescribed treatment for its duration. If $A_i$ is not following it, he will be sanctioned with a lowered reputation on *AVICENA*social network. The smart pill dispenser allows to observe medical prescription adherences, whereas *AVICENA*'s questionnaires allow to observe life-style adherence (*e.g.,* diet and exercise). The

*Norm* $N_1$: Pharmacist $\phi_i \in \oplus$ is obliged to identify patient $A_j \in \mathcal{A}$ and take his prescription $R_k \in \mathcal{R}$ before delivering the medication $M_l \in \mathcal{M}$ to the patient.
*Saction* $S_1$: Competent Authority $C$ sends a warning to pharmacist $\phi_i$ for violating the protocol specified in $N_1$.

| | |
|---|---|
| *Activation Condition* $N_1$ | $hasPrescription(R_k, A_j) \wedge isForMedication(R_k, M_l)$ |
| *Expiration Condition* $N_1$ | $hasDelivered(M_k, \phi_i, A_j)$ |
| *Maintenance Condition* $N_1$ | $True$ |
| *Deadline* $N_1$ | $hasPrescription(\phi_i, R_k) \wedge isIdentifiedBy(A_j, \phi_i)$ |

| | |
|---|---|
| *Activation Condition* $S_1$ | $isViolated(N_1, \phi_i)$ |
| *Expiration Condition* $S_1$ | $warningSent(C, \phi_i)$ |
| *Maintenance Condition* $S_1$ | $True$ |
| *Deadline* $S_1$ | |

Figure 5.8: Formal model for regulative norm $N_1$

*Norm* $N_2$: Patient $A_i \in \mathcal{A}$ has the obligation to follow the prescribed treatment $T_j \in \mathcal{T}$ since the date it starts until the date it finishes. $T_{j'}$ *counts-as* patient's treatment in the context of a patient $A_i$ a treatment starting date $\tau_k \in \mathrm{T}$ and treatment finishing date $\tau_l \in \mathrm{T}$.
*Saction* $S_2$: If patient $A_i$ violates $N_2$ his reputation in *AVICENA*social network is lowered.

| | |
|---|---|
| *Activation Condition* $N_2$ | $isPrescribed(A_i, T_j, \tau_k, \tau_l) \wedge counts\_as(T_j, T_{j'}, A_i, \tau_k, \tau_l) \wedge actualTime(\tau_k)$ |
| *Expiration Condition* $N_2$ | $actualTime(\tau_l)$ |
| *Maintenance Condition* $N_2$ | $followsTreatment(A_i, T_j)$ |
| *Deadline* $N_2$ | |

| | |
|---|---|
| *Activation Condition* $S_2$ | $isViolated(N_2, A_i)$ |
| *Expiration Condition* $S_2$ | $lowerReputation(A_i)$ |
| *Maintenance Condition* $S_2$ | $True$ |
| *Deadline* $S_1$ | |

Figure 5.9: Formal model for regulative norm $N_2$

norm is formally introduced in *Figure* 5.9. Please notice how maintenance condition sets the obligation to maintain a particular state of the world (*e.g.*, perform an action periodically) until another state of the world is met.

Doctors are prohibited from accessing the medical records of patients they do not have assigned. If the doctor access such medical records, he will be sanctioned with an official warning from the competent authority. The norm is formally introduced in *Figure* 5.10. Please notice how prohibitions are modelled as negated states of the world to be maintained while the norm is active. Also, notice the norm activates (*e.g.*, prohibition starts to hold) when the $D_i$ is not assigned to $A_k$, and deactivates (*e.g.*, prohibition does not hold anymore) with the assignation of $D_i$ to $A_k$.

$A_k$ with diabetes visits $D_i$ after a stomach operation. $D_i$ prescribes a low-fiber diet, to

*Norm* $N_3$: Doctor $D_i \in \mathcal{D}$ is prohibited from accessing a medical record $\rho_j \in \mathrm{R}$ if it belongs to a patient $A_k \in \mathcal{A}$ not assigned to him.
*Saction* $S_3$: Competent Authority $C$ sends a warning to the doctor $D_i$ for violating the protocol specified in $N_3$.

| | |
|---|---|
| *Activation Condition* $N_3$ | $\neg isDoctorOf(D_i, A_k)$ |
| *Expiration Condition* $N_3$ | $isMedicalRecordOf(\rho_j, A_k) \wedge isDoctorOf(D_i, A_k)$ |
| *Maintenance Condition* $N_3$ | $\neg AccessMedicalRecord(D_i, \rho_j)$ |
| *Deadline* $N_3$ | |

| | |
|---|---|
| *Activation Condition* $S_3$ | $isViolated(N_3, D_i)$ |
| *Expiration Condition* $S_3$ | $warningSent(C, D_i$ |
| *Maintenance Condition* $S_3$ | $True$ |
| *Deadline* $S_3$ | |

Figure 5.10: Formal model for regulative norm $N_3$

*Norm* $N_4$: Doctor $D_i \in \mathcal{D}$, with the corresponding constitutive power, assigns a new treatment $T_j \in \mathcal{T}$ to patient $A_k \in \mathcal{A}$ with a former treatment assigned $T_l \in \mathcal{T}$. The new treatment is applied for a time period between $\tau_m \in \mathrm{T}$ and $\tau_n \in \mathrm{T}$ whereas the old treatment comprises the period between $\tau_o \in \mathrm{T}$ and $\tau_p \in \mathrm{T}$. The constitutive norm that maps patient's treatment ($T_q$) is updated (dropping the old norm and adding a new one) effectively updating the commitments specified in *Norm* $N_1$.

| | |
|---|---|
| *Activation Condition* $N_4$ | $assignTreatment(D_i, A_k, T_j, \tau_m, \tau_n) \wedge$ $\mathcal{P}(assignTreatment(D_i, A_k, T_j, \tau_m, \tau_n), D_i)$ |
| *Expiration Condition* $N_4$ | $hasTreatment(A_k, T_q, \tau_m, \tau_n) \wedge drop(counts\_as(T_q, T_l, A_k, \tau_o, \tau_p))$ $\wedge counts\_as(T_q, T_j, A_k, \tau_m, \tau_n)$ |
| *Maintenance Condition* $N_4$ | $True$ |

Figure 5.11: Formal model for regulative norm $N_4$

be included in the actual $A_k$'s prescription including medicines, exercise and a sugar free diet. Just like in $S_1$ patient has the obligation to follow the treatment with the same type of sanction to be applied if $A_k$ does not abide. The norm is formally introduced in *Figure* 5.11. Please notice how *constitutive norms* allow for just updating the treatment while the *regulative norm* $S_1$ continues to be responsible for treatment's compliance. Also, notice how constitutive power allows the treatment update to be meaningful, from an institutional point of view. As a final remark, notice how norm is activated when the treatment is assigned, and deactivated when the assignment acquires institutional meaning via the update of the constitutive norm. As this operation is performed by institutional agents (*i.e.* agents controlled by the institution that always will abide norms) the norm can not be violated, therefore there are no sanctions or repair actions applied.

$A_i$ with mild overweight visits doctor. After analysing $A_i$'s medical record, doctor finds out $A_i$'s diet is healthy and overweight is caused by a sedentary style of life. Therefore,

*Norm* $N_5$: Patient $A_i \in \mathcal{A}$ has the recommendation to perform exercise for the time period compressed between $\tau_j \in$ T and $\tau_k \in$ T.
*Saction* $S_5$: If patient $A_i$ exercises his reputation in *AVICENA*social network is increased.

| | |
|---|---|
| *Activation Condition* $N_5$ | $actualTime(\tau_j)$ |
| *Expiration Condition* $N_5$ | $actualTime(\tau_k)$ |
| *Maintenance Condition* $N_5$ | $\neg exercise(A_i)$ |
| *Deadline* $N_5$ | |

| | |
|---|---|
| *Activation Condition* $S_5$ | $isViolated(N_5, A_i)$ |
| *Expiration Condition* $S_5$ | $increaseReputation(A_i)$ |
| *Maintenance Condition* $S_5$ | $True$ |
| *Deadline* $S_5$ | |

Figure 5.12: Formal model for regulative norm $N_5$

doctor decides to recommend weekly routines of exercise to the $A_i$. In order to encourage $A_i$ with following the recommendation, the system will increase $A_i$'s reputation in *AVICENA*'s social network every time it detects (via questionnaires) the $A_i$ has been exercising. The norm is formally introduced in *Figure* 5.12. Please notice how both prohibitions and sanctions can be used to promote desirable pattern of behaviour, in this case following a recommendation. In order to do it, we configure the norm to detect $A_i$'s good practices (in this case, $A_i$ violates the norm when performing healthy actions) and use positive reinforcements (*e.g.*, increasing $A_i$'s reputation) as sanctions.

*Norm* $N_6$: Patient $A_i \in \mathcal{A}$ has the prohibition to consume substances $S_j \in \mathcal{S}$ that are institutionally considered toxic substances.
*Saction* $S_6$: If patient $A_i$ consumes toxic substances the event is logged in his medical record $\rho_k \in$ R.

| | |
|---|---|
| *Activation Condition* $N_6$ | $isPatient(A_i)$ |
| *Expiration Condition* $N_6$ | $false$ |
| *Maintenance Condition* $N_6$ | $counts\_as(S_j, ToxicSubstance) \wedge \neg consume(A_i, S_j)$ |
| *Deadline* $N_6$ | |

| | |
|---|---|
| *Activation Condition* $S_6$ | $isViolated(N_6, consume(A_i, S_j))$ |
| *Expiration Condition* $S_6$ | $actualTime(\tau_m) \wedge isMedicalRecordOf(\rho_k, A_i) \wedge$ $logInRecord(\rho_k, consume(A_i, S_j), \tau_m)$ |
| *Maintenance Condition* $S_6$ | $True$ |
| *Deadline* $S_6$ | |

Figure 5.13: Formal model for regulative norm $N_6$

$A_i$ is elder and suffers from weak liver and is prohibited from consuming toxic substances, such as alcohol and tobacco. If the $A_i$ takes them, considering the high risk of the behaviour, the event is logged in $A_i$'s medical record, and will be presented to $A_i$'s doctor during their next appointment. The norm is formally introduced in *Figure* 5.13. Please notice that the institutional definition of *toxic substance* can evolve through time, but thanks to the abstraction level provided by *constitutive norms* this prohibition will remain the same. Also notice how norm applies to a particular set of patients and never expires.

*Norm* $N_7$: Patient $A_i \in \mathcal{A}$ has the obligation to answer questionnaire $Q_j \in \mathcal{Q}$ presented at time $\tau_k \in$ T before one day.
*Saction* $S_7$: If patient $A_i$ does not answer the questionnaire on time, his reputation is lowered in *AVICENA*'s social network.

| | |
|---|---|
| *Activation Condition* $N_7$ | $questionnairePresented(A_i, Q_j, \tau_k)$ |
| *Expiration Condition* $N_7$ | $actualTime(\tau_l) \wedge hasTimeDifference(\tau_k, \tau_l, OneDay)$ |
| *Maintenance Condition* $N_7$ | $True$ |
| *Deadline* $N_7$ | $questionnaireAnswered(A_i, Q_j)$ |

| | |
|---|---|
| *Activation Condition* $S_7$ | $isViolated(N_7, A_i)$ |
| *Expiration Condition* $S_7$ | $lowerReputation(A_i)$ |
| *Maintenance Condition* $S_7$ | $True$ |
| *Deadline* $S_7$ | |

Figure 5.14: Formal model for regulative norm $N_7$

A new $A_i$ joining the *AVICENA* program acquires the obligation to answer the questionnaires related to the treatment program recommended by his doctor. Patients have one day to fill the questionnaires, but this time period can be easily tailored according to scenario requirements. The norm is formally introduced in *Figure* 5.14. The spirit of this norm is trying to improve system's health monitoring capabilities by encouraging users to fill the questionnaires via a gamification approach [McC12]. The idea is making questionnaires not only user friendly, but also providing a social reward for filling them. If patient does not fill the questionnaire in the time period established, his reputation in *AVICENA*'s social network is reduced.

### 5.4.2 Norm operations

Once we have introduced formal examples of the norms we can proceed with examples of how norm management operations are affecting the different set of norms formally defined in our scenario. This subsection presents example of norm operations based on normative environment modifications. Every example consists in an informal explanation of the operation in natural language which focuses on the legal implications of the operation performed. The example also includes a formal representation of the operation using time lines.

As seen in *Figure* 5.15, when norm $N_5$ is promulgated retroactively patient $A_i$ will be rewarded for exercising (please, remember the sanction of this norm consists in a reward tailored to promote healthy habits). In order to maximize the effect of encouraging a positive behaviour, the patient is rewarded both for exercising after the norm is in place (time $\tau_3$) and also before it is (times $\tau_1$ and $\tau_2$).

As depicted in *Figure* 5.16, when patient is assigned a treatment requiring life style adherence, norm $N_7$ is promulgated prospectively. From this point, patient $A_i$ acquires

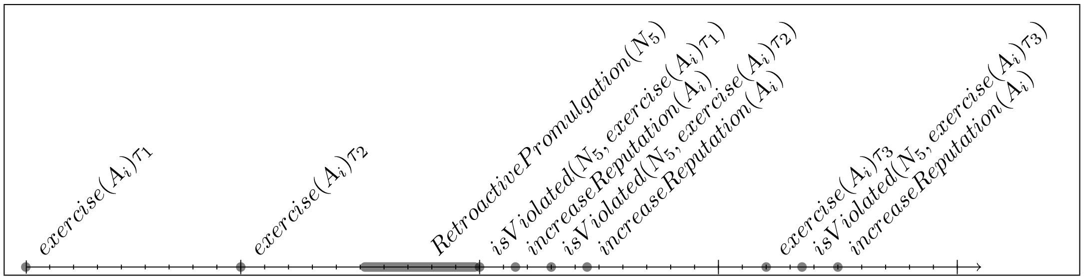



Figure 5.15: Retroactive Promulgation of regulative norm $N_5$

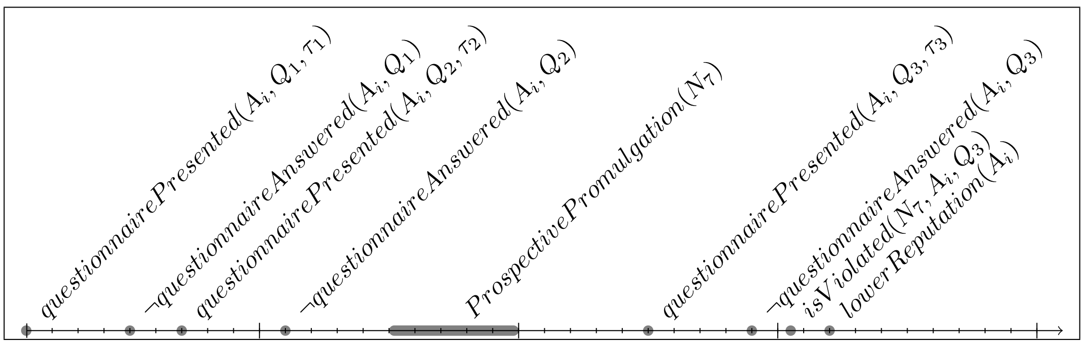



Figure 5.16: Prospective Promulgation of regulative norm $N_7$

the obligation to fill questionnaires $Q_1$, $Q_2$ and $Q_3$. Therefore $A_i$ will be sanctioned for not filling them. Questionnaires not filled before the promulgation (times $\tau_1$ and $\tau_2$) will not be sanctioned (*e.g.,* we can consider them to be optional questionnaires for statistical purposes) but missing a questionnaire after the promulgation (time $\tau_3$) implies a lowered reputation in the *AVICENA* social network, because life style adherence control is required for the correct treatment.

As seen in *Figure* 5.17, according to questionnaires, patient's life style adherence to current treatment is perfect, as he has been following every recommendation timely. As a reward, his doctor decides to remove the obligation to fill questionnaires. The patient does not need *AVICENA* questionnaires anymore to control part of his treatment adherence. Therefore patient's lifestyle adherence will be controlled by scheduled medical analytics and visits to the doctor to control his evolution. Norm $N_7$ is annulled. From this point patient $A_i$ can miss questionnaires (*e.g.,* $Q_1$, $Q_2$ and $Q_3$) without a sanction (time $\tau_3$). In case he missed a questionnaire and was sanctioned for this (time $\tau_1$) the sanction is undone (*i.e.* patient's reputation is restored). In case he missed a questionnaire and the sanction is still pending (time $\tau_2$), the sanction will not be applied.

*Figure* 5.18 shows an example of an e-Prescription system implemented in the country where the patient lives. As the e-Prescription information systems connects directly $\mathcal{A}$, $\mathcal{D}$

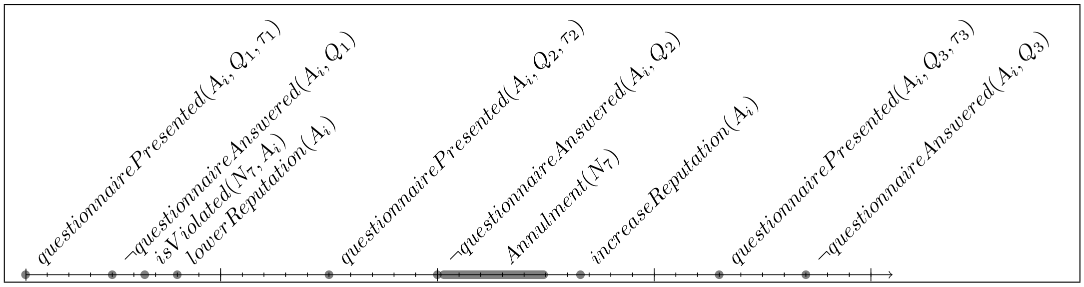



Figure 5.17: Annulment of regulative norm $N_7$

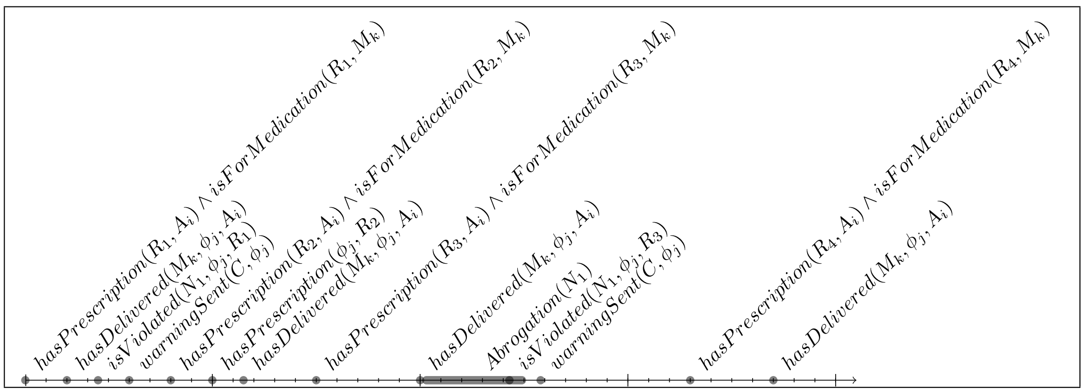



Figure 5.18: Abrogation of regulative norm $N_1$

and $\oplus$, the obligation of pharmacist to identify patients and take their prescription before delivering medication is not meaningful anymore. Therefore norm $N_1$ is abrogated. For simplicity, we will focus in the obligation of pharmacist $\phi_j$ to take the prescriptions (*e.g.*, $R_1$, $R_2$, $R_3$ and $R_4$) during the example. The obligation to identify the patient can be easily extended from this example. From this point, pharmacies can deliver medication without taking the prescription, and not be sanctioned for this (prescription $R_4$). In case the pharmacy was sanctioned for delivering medicines without taking the prescription in the past (prescription $R_1$) the sanction is kept. In case the pharmacy delivered medicines without prescription (prescription $R_3$) and the sanction is still pending, the sanction will be applied after the abrogation.

As seen in *Figure* 5.19 when patient treatment is updated by doctor with the corresponding constitutive power (*e.g.*, prescribing a low fibre diet after a stomach operation) the constitutive norm accounting for the patient treatment is updated prospectively. In this particular example, norm $N_4$ is promulgated prospectively, and it affects indirectly norm $N_2$, as it uses the constitutive norm in the obligation. Therefore, patient $A_i$ will be sanctioned for taking fibre. If the patient took fibre before the promulgation (times $\tau_1$ and $\tau_2$) he will not be sanctioned (*e.g.*, because his institutional interpretation of the treatment

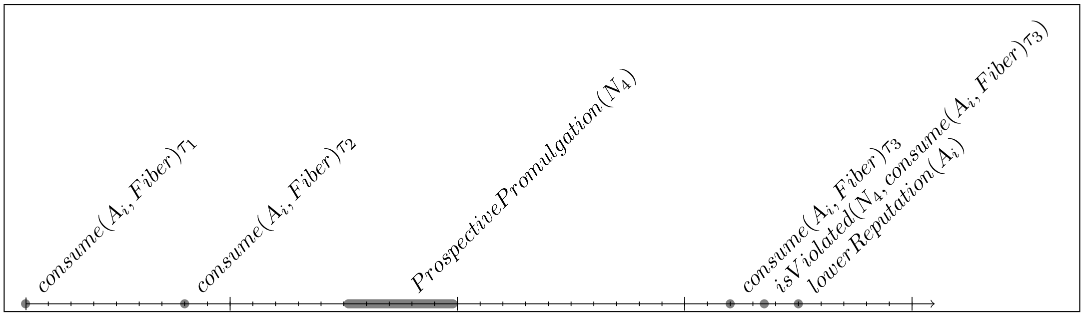



Figure 5.19: Prospective Promulgation of regulative norm $N_4$

did not restrict fibre back then) but taking fibre after the promulgation (time $\tau_3$) implies a lowered reputation in the *AVICENA* social network (according to $N_2$ which has been indirectly affected), because patient's current treatment is restricting fibre consumption.

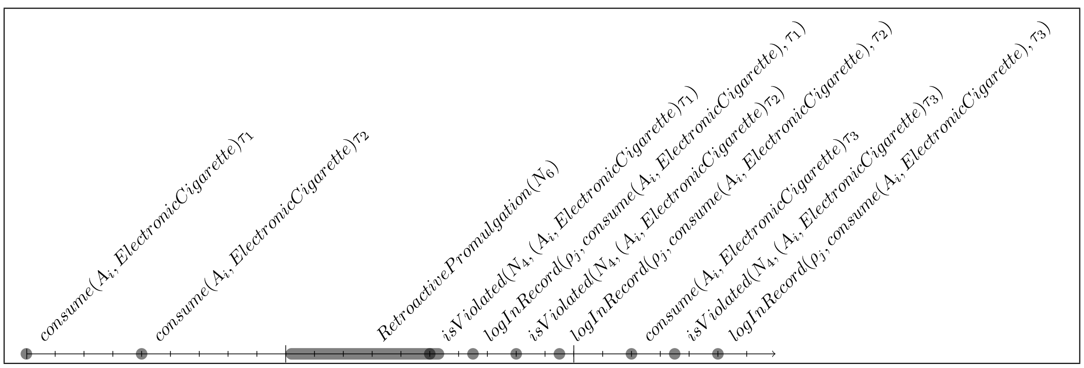



Figure 5.20: Retroactive Promulgation of regulative norm $N_6$

As depicted in *Figure* 5.20 patient's treatment can also suffer updates due to technological changes. Furthermore, the updates can also be retroactive. In this particular example, norm $N_6$ is promulgated retroactively, effectively updating the definition of toxic substance due to a technological change: the popularization of the electronic cigarettes. Therefore, patient $A_i$ will be sanctioned for smoking (*i.e.* technically consuming) electronic cigarettes. If the patient smoked electronic cigarettes before the promulgation (times $\tau_1$ and $\tau_2$) he will be sanctioned (*e.g.*, because the institutional interpretation of toxic substance is updated with retroactive effects) just as if he smoked them after the promulgation (time $\tau_3$). Please notice the nature of the sanction (*i.e.* logging the event in patient's medical record $\rho_j$) allows the retroactive promulgation to dynamically generate a medical record adapted to the changes in the institutional interpretations. The valuable information in such medical record will allow doctor to asses patient's behaviour with a holistic view, taking into account not only the actions performed by the patient after the update in the institutional interpretation, but also the previous actions.

In the realistic scenario described in §5.1.3 Lear is an elder patient with 3 daughters who lives 4 months of the year with each one of them. Their daughters live in three different cities of two different European countries, therefore, patient lives 4 months in every city. Every time the patient changes his residence, he is assigned a new doctor. The assignment of a new doctor implies an update on the constitutive powers assigned to the doctor. In the scenario, patient $A_i$ moves from Spain where doctor $D_{BCN}$ is assigned to him, to France where he is assigned to doctor $D_{PAR}$. Therefore, doctor's $D_{BCN}$ constitutive powers are abrogated. Treatments prescribed by doctor $D_{BCN}$ to patient $A_i$ in the past are valid (and the patient has the obligation to follow them even after the abrogation), but treatments prescribed after the abrogation have no institutional effect. Once doctor$D_{PAR}$ is assigned to patient $A_i$ his constitutive powers are promulgated prospectively. That is, treatments prescribed by doctor $D_{PAR}$ to patient $A_i$ before the promulgation have no institutional meaning. However, treatments prescribed after the abrogation have full institutional effect, and therefore, the patient has the obligation to follow them.

## 5.5 Scenario implementation

Once we have introduced our theoretical framework for the testing scenario we proceed to depict the implementation performed for our testing framework. We will focus on depicting the architecture and the set of norms resulting from simulating the different cases in our testing scenario.

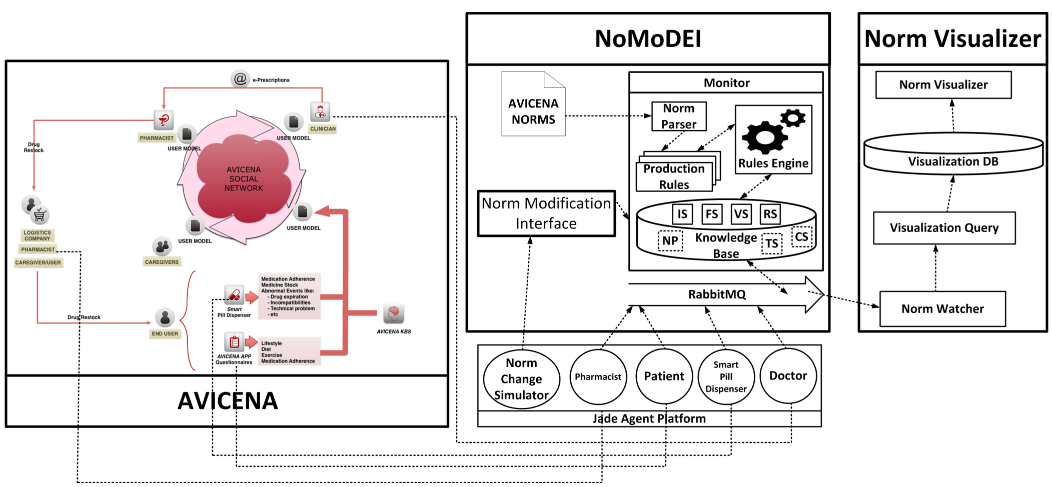



Figure 5.21: Architecture for our testing scenario

As depicted in *Figure* 5.21 we have performed the implementation by simulating the behaviour of the different actors involved in the *AVICENA* platform (including human actors such as pharmacist and artificial actors such as the pill dispenser) in an agent platform. We have used the JADE agent platform [BPR01] because it allows to easily develop and deploy agents based on the Java programming language. The platform is FIPA-compliant and has some interesting ad-ons (such as the mental inspector and the sniffer) that help during agent development and deployment. Furthermore, we already have some experience using the platform. The agents in the platform perform tasks associated to the actors in the scenario they simulate. The tasks are passed through the event bus to the NoMoDEI platform where the monitor puts them in contrast with the set of norms in the scenario. A static program triggers norm changes that have prospective or retroactive impact on the normative state of the system. Finally, we use a norm visualizer to check the impact of agent actions and norm operations in the scenario.

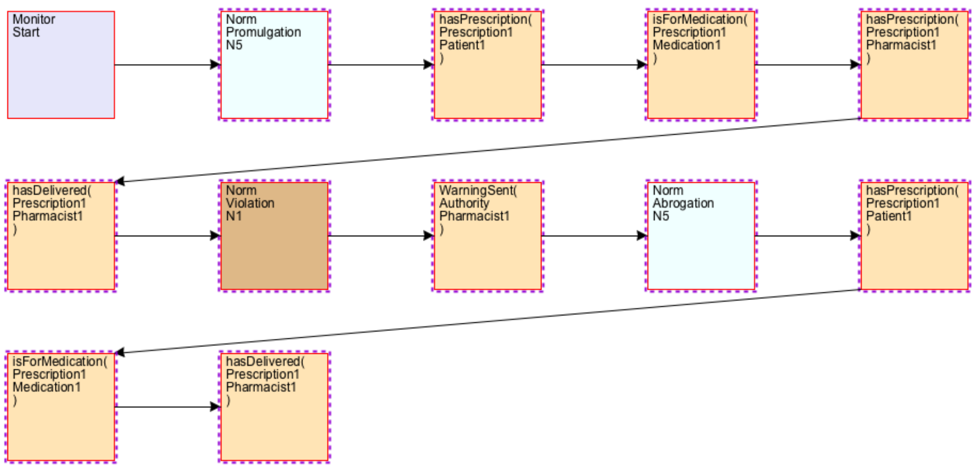



Figure 5.22: Abrogation of a norm: timeline

*Figures* 5.22 and 5.23 show the abrogation of a norm in our test scenario. The example corresponds to *Norm* 5.9 and time-line 5.18 in §5.4. *Figures* 5.24 and 5.25 show the retroactive promulgation of a norm in our test scenario. The example corresponds to *Norm* 5.12 in §5.4. Finally, *figures* 5.26 and 5.27 show the retroactive promulgation of a constitutive norm in our test scenario. The example corresponds to *Norm* 5.13 and time-line 5.20 in §5.4.

Via our testing implementation we have demonstrated we can integrate agents in a domain (*i.e.* the medical domain) which is subject to tight regulations and protocols. Agent interactions are observed and put in contrast with the regulations regimenting the scenario, therefore we can state the agent system is subject to such regulations. Furthermore, regulations and protocols evolve through time, and the information system should be able to reflect this evolution. Therefore, our implementation is flexible and it shows how Normative Systems applied to e-health scenarios can effectively adapt to new regulations and protocols caused by social, technological, medical and contextual changes.

## 5.6 Conclusions and Related work

Once we have finished introducing our approach of a norm-aware agent-based model for integrated Wastewater management systems, we proceed to put it in contrast with similar approaches. This section compares the state of the art analysed in §2.4 with the work


Figure 5.23: Abrogation of a norm: norm events

presented in this *Chapter*. We will provide a brief summary of every proposal analysed in §2.4 and compare it to our proposal, focusing on detecting confluence points where our work can be complemented by the different approaches in the state of the art. Then, we will outline conclusions.

### 5.6.1 CARREL

In §2.4.1 we analyse the CARREL and CARREL+ frameworks, virtual organization formalized using ISLANDER [EDLCS02] and focused on procuring organs and tissues for transplant. Both CARREL and CARREL+ apply Electronic Institutions for ensuring the *safety* and *soundness* of the resulting system. This effectively allows to create systems that are compliant with a particular set of norms and regulations. However, CARREL and

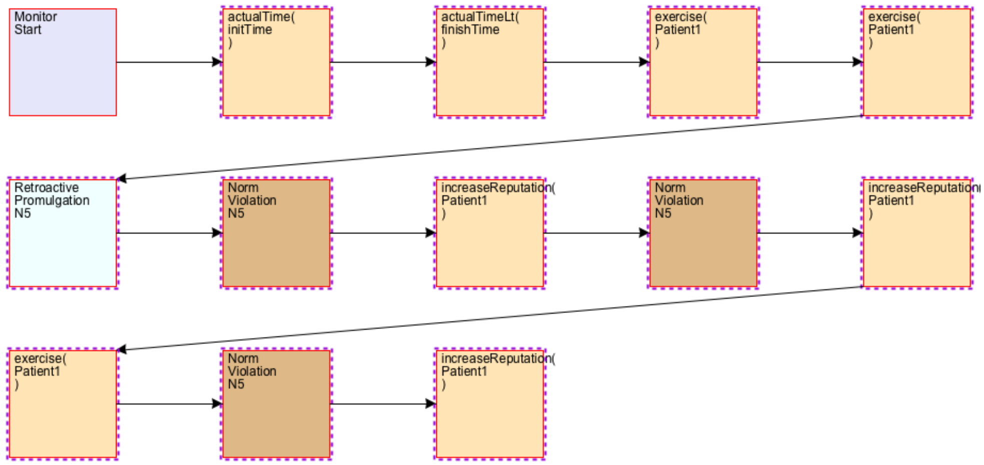



Figure 5.24: Retroactive promulgation of a norm: timeline

CARREL+ present three main differences with the system we propose:

- Communicative acts (in the case of CARREL) and arguments (in the case of CARREL+) that do not abide to the norms and protocols specified are directly rejected. There are no sanctions or repair actions associated to a norm or protocol violation because they are never violated. This fact is limiting agent's autonomy, as agents do not have the choice to violate a norm or protocol in case they consider it would be beneficial for the society. For instance a doctor might access a restricted patient medical record in case patient's life is at risk, and there is no other doctor available. Furthermore, it is also limiting norm's expressiveness, as our system allows to define norms that can be violated (mild sanctions) norms that should be never be violated (hard sanctions with repair actions associated) and even promote norm violation by applying positive sanctions (*e.g.,* increasing patient's reputation on the social network when he does physical exercise).
- Norms and protocols are applied by limiting communicative acts. This makes it impossible to observe and control the behaviour of agents which are not acting through the exchange of communicative acts. It will typically happen in scenarios where we have both computational and human agents interacting together in the institution. In such cases, CARREL is not able to observe and enforce the behaviour of human agents, whereas our system is capable of doing it.
- CARREL and CARREL+ can adapt to changes in regulations and protocols, just like the system we propose. However, our system is able to perform this task at runtime, without stopping the observation of the institutional reality (*e.g.,* which norms have been violated), and inferring an institutional reality that is consistent with the updates performed on the set of norms (*e.g.,* if a norm is removed, we can remove the sanctions associated to the norm). It is unclear if CARREL and CARREL+ support these features.

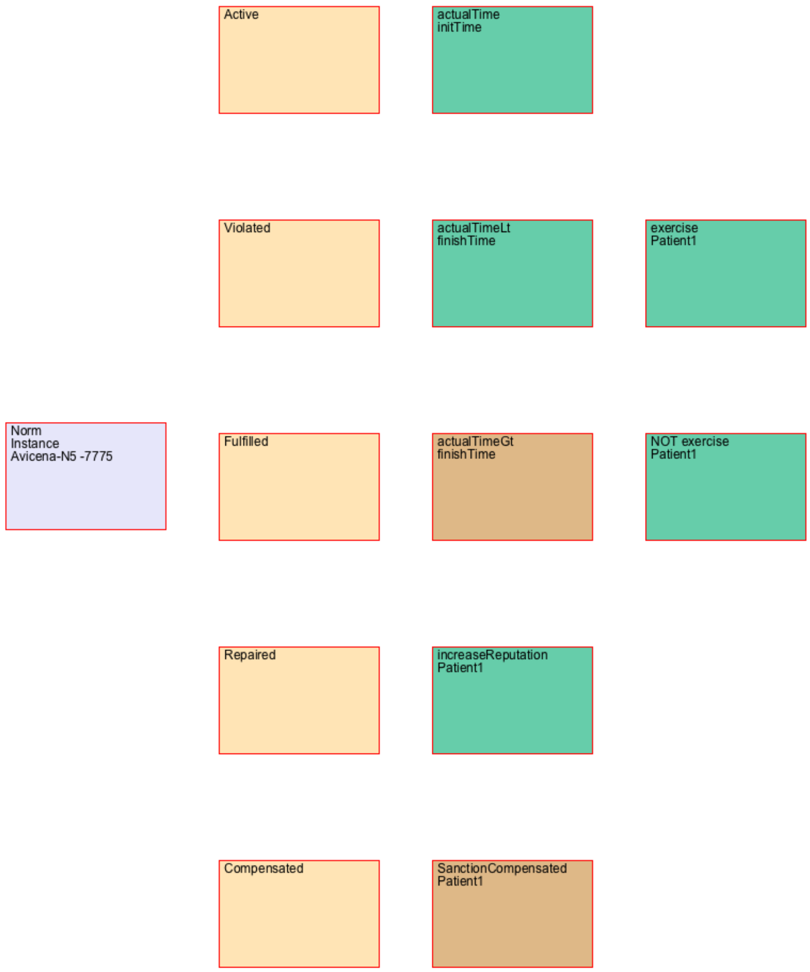



Figure 5.25: Retroactive promulgation of a norm: norm events

### 5.6.2 Automated monitoring of medical protocols

In §2.4.2 we analysed a system for automated real-time monitoring of medical protocols. This framework presents many similarities with the one analysed in § 2.4.1. Communicative acts that do not abide to norms and protocols are rejected limiting both agent's autonomy and norm's expressiveness. Furthermore, it is impossible to observe and control the behaviour of agents (*e.g.*, human agents) which are not acting through the exchange of communicative acts. Moreover, it is unclear how the framework presented in [AAB$^+$03] will support norm dynamics (*i.e.* updating the set of norms regimenting the institution). Finally, the architecture presented is based on a knowledge distributed approach that allows to distribute the load among agents, effectively defining a dynamically scalable system. It is clear the monitoring load is distributed among the set of *supervisor agents* in the system,

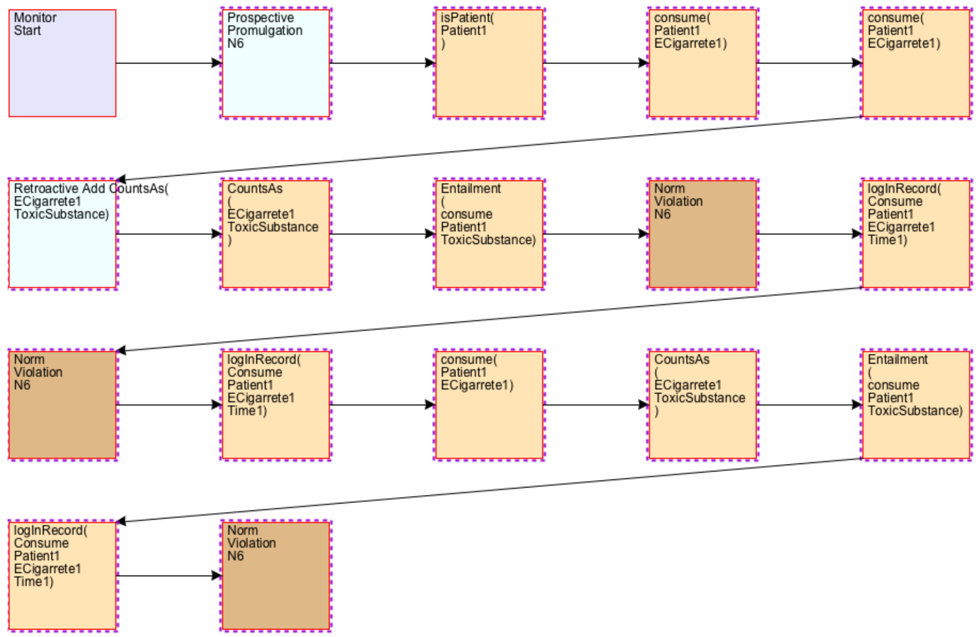



Figure 5.26: Retroactive promulgation of a constitutive norm: timeline

but the criteria followed for load distribution is unclear. We have also explored the option to distribute monitoring load among distributed computational resources [GSÁNVSF12] but in our case we provide a distribution criteria based on a formal analysis of norm dependencies.

### 5.6.3 A Multi-Agent System approach for monitoring the prescription of restricted use antibiotics

In §2.4.3 we analysed a Multi-Agent System to monitor and help in the revision of medical prescriptions including antibiotics of restricted use. Once again, the framework analysed in this subsection presents many similarities with the one analysed in §2.4.1. Agent actions that do not abide to norms and protocols are filtered by the *governor* agents, effectively limiting both agent's autonomy and norm's expressiveness. Furthermore, this work proposes attaching agents to human actors in order to integrate them in the Multi-Agent Systems, whereas our proposal opts to fully integrate human actors in the institution, effectively supporting institutions formed both by human and computational agents. Finally, it is also unclear how the framework presented in [GPGS+03] will support norm dynamics (*i.e.* updating the set of norms regimenting the institution).

### 5.6.4 Toward a Conceptual Agent-based Framework for Modelling and Simulation of Distributed Healthcare Delivery Systems

In §2.4.4 we analyse the $AOE^2$ framework for agent-based modelling and simulation in distributed healthcare delivery systems. The idea of applying a framework to agent-based

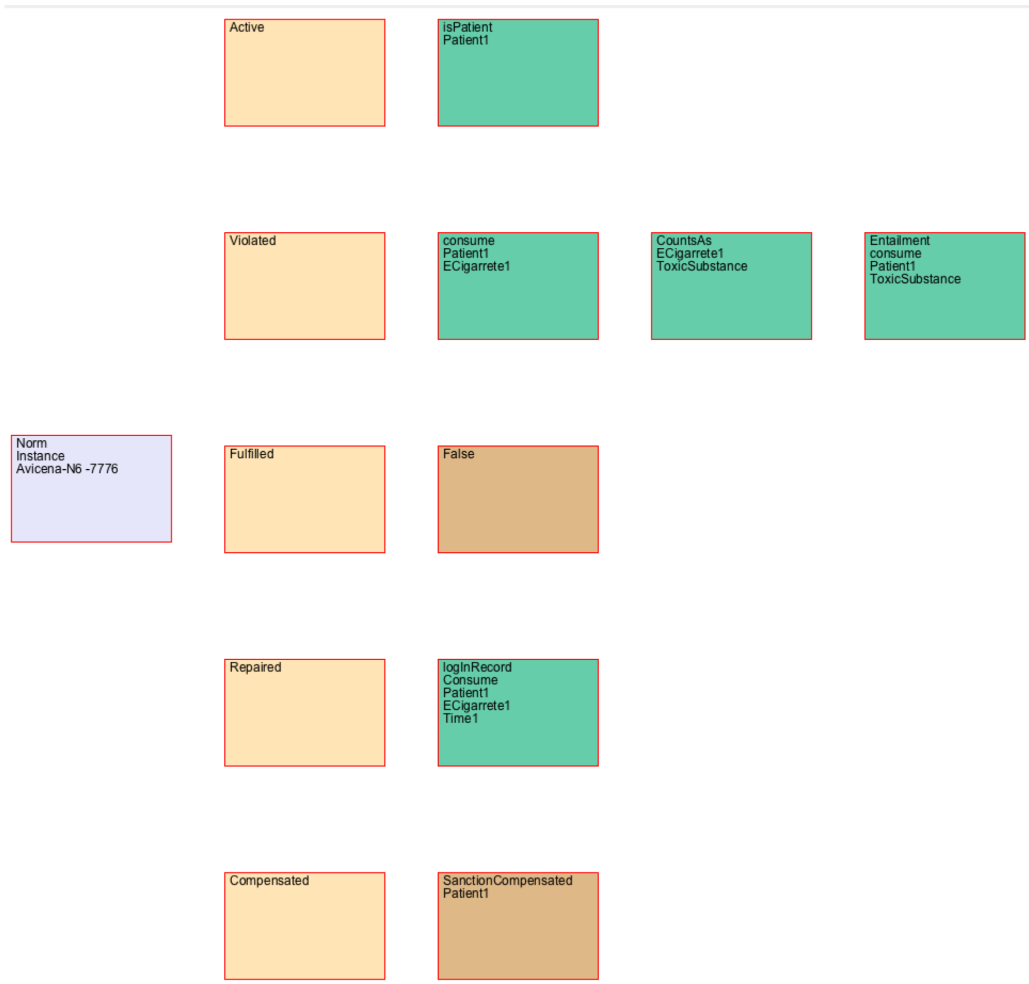



Figure 5.27: Retroactive promulgation of a constitutive norm: norm events

simulations in the healthcare domain is appealing. However, among the set of elements and concepts the $AOE^2$ framework is taking into account we miss one that is highly important in regulated scenarios such as the healthcare domain: Norms, that in the form of obligations, permissions and protocols model the expected patterns of behaviour to be followed by the different actors involved in the scenario. One can easily envision the multiple problems policy makers involved in healthcare will find when trying to balance the norms and protocols between an efficient system (with relaxed norms tailored to minimize overhead) and a safe one (with strict norms designed to minimize deviance from the expected behaviour). Being able to model (and apply in agents' behaviour) such norms will effectively allow an agent-based simulation to test the outcome of modifying the normative environment (that is, the set of norms) in the institution represented in the simulation (*e.g.*, hospital or coordinated health care system in a particular area). We consider a system like the $AOE^2$ framework could benefit from the enhancement provided by our normative model and our system for updating the normative environment at runtime.

### 5.6.5 Conclusions

This *Chapter* has presented *AVICENA* an extension of the COAALAS project , a framework for Multi-Agent Systems that combines organisational and normative theories with Ambient Assisted Living (AAL) technologies. The COAALAS project has been extended with a dynamic Normative System grounded on the ALIVE framework that allows us to create a computational model for the regulations and protocols around treatment prescription and treatment adherence.

The *AVICENA* approach and its tools, as they are envisaged, belong to a new generation of intelligent services that are designed to be safely deployed and used by patients alongside other assistive devices to support their owners in their daily life and improve their concordance, adherence and persistence to a therapy. Consequently this will augment their wellbeing, support concordance and interaction with their caregivers and doctor. In this sense, the *AVICENA* platform will be a powerful extender of user's capabilities and serve society by reducing care costs and providing valuable knowledge about the everyday people's experiences in dealing with a therapeutic regime. It is important to highlight that one of the crucial factors for the success of the *AVICENA* approach is stepping away from vertically-oriented, closed systems towards the use of open systems, based on open APIs and standardized protocols, as the ones provided by COAALAS, at various system levels.

Through the *ALIVE* framework COAALAS provides a flexible multi-level architecture able to model the complex interactions among different actors involved in the AAL tasks, where a set of heterogeneous actors have different responsibilities and offer or consume different services. Thanks to *ALIVE*'s multi-level approach, COAALAS supports introducing changes at a high level (*e.g.*, introducing a new actor, a new objective for the system or a new expected pattern of behaviour) without performing any modification to the lower levels (*i.e.*, reprogramming the agents in the different smart devices) because changes at higher levels automatically trigger changes at the lower levels. COAALAS provides AT devices with a high level layer, easy to use and understand by non-technological experts. By using this layer, AT devices can be easily adapted to introduce a new medication regime or to include a new actor (*e.g.*, patient's relative as a caretaker) in the system.

Also via the *ALIVE* framework, COAALAS allows for monitoring the different actions performed by the set of actors in order to fulfill the AAL tasks. Deviations from the expected patterns of behaviour can be detected, and sanctions (when applicable) or repair actions can be applied. Therefore, *AVICENA* is able to provide the appropriate support for dealing with unexpected events (*e.g.*, sending a doctor, or an urgent shipment of medications to the patient when the pill dispenser device runs out of pills).

Finally, the *ALIVE* framework provides COAALAS with a set of intelligent agents that support both exception handling and organisational normative awareness capabilities. Exception handling is common in other service-oriented architectures, however, most approaches tend to focus on low-level (*i.e.*, services provided by the actors) exception handling. The *ALIVE* approach enables managing of exceptions at multiple levels ranging from substituting services (*i.e.* low level exception handling) to looking for alternative ways to achieve a particular goal (*i.e.*, high level exception handling). Regarding organisational normative awareness, making normative agents reason about their tasks before performing them, and discarding the ones that do not comply with the expected patterns of behaviour, adds organisational awareness to the execution of the different tasks.

AT are applied to support people in their daily life. Most approaches focus solely on the direct interaction between users and the assistive tool [KBLA+09] but *AVICENA* is a more complex environment where more heterogeneous interactions are permitted. AI has the potential to provide innovative mechanisms and methods capable of taking into account more complex interactions. For instance, the important role that third parties may have in user activities, and explicitly reflect the social constraints that apply in the relationship between device and patient. For instance, a simple reminder system can be implemented using a smart-phone's calendar and alarm systems. However, it would lack the autonomy, social awareness and normative awareness our proposal provides. A simple alarm system is not able to adapt reminders to user's calendar (it will keep reminding the user to take a medication dose even if his calendar indicates he will be away from home when it is time to take his medication, rather than adapting the reminder to user's schedule) nor is able to alert caretakers if potentially dangerous deviations from user's routine are detected.

Our approach focuses on making devices intelligent enough to organize, reorganize and interact with other actors providing smart devices with an awareness of their social role in the system (including commitments and responsibilities). This way, smart devices are capable of reacting to deviations from the expected patterns of behaviour, effectively adapting to a wide range of AAL situations that could have an impact on the well-being of the user. This happens to be a very useful tool for the *AVICENA* approach.

The particularities of each elder's disabilities makes any custom solution difficulty exportable to a whole population of elders. Furthermore, if the characteristics of a particular elder's disability change over time (*e.g.*, the disability is degenerative) the applicability of a custom solution is also limited in time. In order to tackle these issues, we present COAALAS a system based on an adaptable and extensible architecture. By using COAALAS elder-care experts can easily *adapt* the system capabilities to the patient's current state. What is more the extensibility of COAALAS (through the addition of more agents and services) provides a virtually endless amount of tools for elder's support.

Our proposal is specially relevant when applied to scenarios that are not only regulated, but also contain dynamic regulations. That is, the normative environment to be taken into account (*i.e.* the medical regulations and protocols to be applied) is not static, but changes over time as new regulations arise and obsolete protocols are replaced by modern ones. Changes in normative environments are typically caused by social changes or technological changes. Some examples of such changes are:

- Patient's treatment is updated: Typically, after a visit with his doctor, the patient might have an updated prescription, depending on his evolution and medical history. Therefore, the patient has a new set of obligations and permissions.
- Medical advances: New studies demonstrate a particular habit is non-healthy, and patient's treatment is updated.
- Technological advances: New technologies (*e.g.,* electronic cigarettes, e-prescriptions, *etc.* ) will typically imply an update on medical regulations and protocols.
- Social changes: Habits that were widely accepted in the past (*e.g.,* self-prescription of medications) are not socially accepted anymore.
- Environmental changes: Agent actions can alter the environment, and therefore change the set of norms regulating the system. For example, a patient living in dif-

ferent countries along the year will be subject to different medical regulations. The transition between the different regulations requires a dynamic Normative System.

In order to support such a highly regulated and dynamic scenario we propose a Normative System and integrate it into the *AVICENA* framework. The main contributions of the proposed system *w.r.t.* related work (as presented in §2.4) is proposing a system that meets the following properties:

- **Flexible**: Actors have the flexibility to violate norms (*i.e.* regulations and protocols). When such norms are violated both sanctions (discouraging the agent from following non-complying behaviours in the future) and repair actions (aimed at bringing the system back from an undesirable state to a normal state) are applied. On the one hand it allows us to easily integrate both human and computational actors, as there is no feasible procedure to fully prevent human actors from violating a norm. On the other hand, the level of the sanction applied when violations are detected allows us to define a wide range of norm compliance modes such as:
  - Norms that should never be violated. For instance, it is forbidden to self-prescribe medications. Violating such norms will typically imply a very hard sanction and a repair action.
  - Norms that can be violated in very exceptional cases. For instance, a doctor can only access his patient's medical records, but in case patient's life is at risk and no other doctor is available, a doctor might access the medical records of a patient not assigned to him.
  - Norms that can be violated occasionally: For instance, a patient that is on diet and skips it once a month might remain unsanctioned.
  - Norms that should be violated: By applying positive reinforcement as sanctions, we can effectively detect healthy patterns of behaviour and promote them.
- **Expressive**: We support a wide range of normative concepts. First, deontic norms, including obligations, permissions and prohibitions. Second, constitutive norms, allowing to interpret brute facts from an institutional point of view. This allows abstracting deontic norms effectively providing simple and clear normative contexts. Finally, constitutive powers, allowing to provide institutional meaning to actor actions performed under particular circumstances.
- **Dynamic**: We support norm dynamics, effectively allowing to insert, remove or update norms in the normative context. The normative context can be updated at run-time, while the system is checking if regulations and protocols are abided, and inferring a normative state [UBSA10] consistent with the updates performed (*e.g.*, removing an obligation can remove the sanctions associated to previous violations of the obligation).

We consider our proposal is a relevant contribution to the state of the art. The medical domain is subject to tight regulations and protocols. Therefore the information systems deployed on the medical domain should be subject to such regulations. However, in order to be able to react to unexpected and potentially dangerous situations, informations systems must be flexible, accounting for violations on the regulations and protocols. For instance, a doctor should be able to access a restricted medical record of a patient (at least, partially) if the patient's life is at risk and no other doctor is available. Furthermore, medical regulations and protocols evolve through time, and the information system should be able to reflect this evolution. Therefore, our proposal is relevant, as it is flexible and it allows Normative Systems applied to e-health scenarios to adapt to new regulations and protocols caused by social, technological, medical and contextual changes.

In contrast, the related work presents an interesting approach, specially the $AOE^2$ presented in §2.4.4. Combining our proposal with the $AOE^2$ framework would allow agent-based simulations to test the outcome of modifying the normative environment in health care domains. This is very interesting, as such system would allow legislators to balance the norms and protocols between an efficient system with relaxed norms and a safe system with strict norms.

In this chapter we have seen a more specific set of norms and protocols, when compared to §4. The chapter has two main focus. First, how constitutive norm dynamics can be used to design a clear system, easy to maintain and update. For instance, updating medical treatments after visiting a doctor. Second, how constitutive powers add security to the system and can be effectively applied in multi-institutional settings. The norms presented tackle the following tasks:

- Detect and solve in a timely manner potentially dangerous situations, such as a patient not taking his medication.
- Control security issues related to medical data, such as an unauthorized doctor accessing a patient medical record.
- Control medical protocols, such as preventing self-medication.

As a summary, the main contributions of this chapter are:

1. A model of the agents involved in the e-health scenario.
   a) The model includes a social structure developed using the ALIVE [APV+10] methodology.
   b) The model includes a normative structure specified in the norm formalism we use on this document.
2. An instantiation of the generic architecture presented in §3 to the e-health scenario presented in this *Chapter* .
3. Implementation details and tests on the architecture.
4. The proposal in this *Chapter* establishes the basis for secure agent-based systems for e-health. In this scope, security is understood as the agent's ability to be norm-aware, effectively abiding to the medical regulations and protocols.

The *AVICENA* scenario has been kindly provided by Doctor Cristian Barrué. The COAALAS project has been developed in collaboration with Sergio Álvarez-Napagao, Dario Garcia-Gasulla and Jonathan Moreno.

CHAPTER 6

# Conclusions

This thesis focuses on the formal definition and development of a normative monitoring framework and run-time architecture to support scenarios where the normative context is dynamic and may evolve through time. The motivation for this work is that in most real domains human regulations are not static but dynamic, they change in order to adapt to new situations and behaviours. However, current work in literature on automatic normative monitoring tends to assume static sets of norms, and therefore a regulatory change would require to stop he monitoring processing, amend the normative specification used, and the resume the task, with the risk to stop observing meaningful (potentially harmful) events during the monitor down-time.

In order to fulfil our main objective we introduce (in §3.3) a basic formal framework for norm monitoring, which then we extend (in §3.4) accounting for norm change. Then (in §3.5) we demonstrate we can perform norm-update operations on the fly, without having to stop the monitoring process. We accomplish this thanks to an auxiliary monitoring process that helps with the norm update and then is merged with the main monitoring process. We meet our objective by providing not only formal definitions, but also architectural definitions and a working prototype.

Furthermore, our algorithms also show the monitoring process is inferring a normative state consistent with the norm change. For instance, previous sanctions are maintained in abrogated norms and compensated in annulled norms. Specifically, we support four context update operations that have a consistent impact on the normative context, mainly:

- Prospective Promulgation: Where a norm is inserted in the system. It can be activated but not violated by past events.
- Retroactive Promulgation: Where a norm is inserted in the system. It can be both activated and violated by past events.
- Annulment: Where a norm is removed from the system. Norm violations are removed, and agents that paid a fine for violating the norm receive a compensation. Repair actions are un-done when this is relevant (*e.g.,* removing a ban from an actor for violating the norm).

- Abrogation: Where a norm is removed from the system. The norm is marked as being *'in transition'* and can not be violated anymore. However, violated and activated instances of the norm remain valid.

Another important objective in our work was to support a norm formalisation that is both expressive and flexible. We support a wide range of normative concepts including obligations, prohibitions, permissions, constitutive norms and institutional powers, including constitutive powers and normative powers. We also support a wide range of elements inside our normative concepts, including activation, maintenance and deactivation conditions, as well as deadlines. Finally, we support up to four norm update operations giving the legislator or policy maker a wide range of options for creating the norms and maintaining them. Even though some of the options we provide are explicitly invalid in the real world (*e.g.,* retroactive promulgation is prohibited by the constitutions and bills of rights in many countries) we provide the option for the sake of flexibility. We show both the expressiveness and flexibility of our framework by applying it to two real world scenarios in §4 and §5.

As a summary, the main results and contributions of each chapter are:

1. **Chapter 3**
   a) A conceptualization of a norm monitoring framework supporting the context update operations.
      i. We include four operations supporting prospective and retroactive norm insertion and deletion. We also define norm updates.
      ii. The operations can be applied to regulative and constitutive norms.
      iii. The conceptualization is enhanced to support constitutive powers and normative powers.
   b) A formal model of the norm monitoring framework, including:
      i. A formal model of the four norm update operations.
      ii. Formal algorithms for supporting the operations.
      iii. Extension of the formalism, so the operations can support both regulative and constitutive norms. The extension is also supported via formal algorithms.
      iv. Extension of the formalism to include constitutive powers and normative powers. The extension is also supported via formal algorithms.
   c) A base architecture for the norm monitoring framework.
   d) An extension of the base architecture to support norm update operations, constitutive powers and normative powers.
   e) An implementation of the architecture.
   f) A prototype for a norm visualization component for documenting the results of our research.
2. **Chapter 4**
   a) A model of the agents involved in wastewater management in the river basin.
      i. The model includes a social structure developed using the ALIVE [APV$^+$10] methodology.
      ii. The model includes a normative structure specified in the norm formalism we use on this document.
   b) An instantiation of the generic architecture presented in §3 to the wastewater management scenario presented in this *Chapter* .

   c) Implementation details and tests on the architecture.
   d) The proposal in §4 establishes the basis for performing agent-based social-aware simulations in the river basin scenario.
3. **Chapter 5**
   a) A model of the agents involved in the E-health scenario.
      i. The model includes a social structure developed using the ALIVE [APV+10] methodology.
      ii. The model includes a normative structure specified in the norm formalism we use on this document.
   b) An instantiation of the generic architecture presented in §3 to the E-health scenario presented in this *Chapter* .
   c) Implementation details and tests on the architecture.
   d) The proposal in §5 establishes the basis for secure agent-based systems for E-health. In this scope, security is understood as the agent's ability to be norm-aware, effectively abiding to the medical regulations and protocols.

## 6.1 Relevance to Artificial Intelligence

Multi-Agent systems are a central domain in Artificial Intelligence, evolved from Parallel Artificial Intelligence, Distributed Problem Solving and Distributed Artificial Intelligence. Normative Systems are a relevant domain in Artificial Intelligence too. They apply social abstractions in order to allow complex multi-agent systems to cope with coordination co-operation and trust issues. In order to apply such social abstractions, it is often mandatory to know the current state of the normative environment. However, inferring this state can prove to be a challenging problem either due to having to deal with dynamic normative contexts, which are common in human regulation systems and real world scenarios.

This proposal aims to tackle this challenge. Therefore, it is tightly related to relevant domains in Artificial Intelligence such as Normative Systems and social order in Multi-Agent systems. The aim of this proposal might be seen as allowing commonly accepted Artificial Intelligence techniques such as Normative Systems and Multi-agent systems to be potentially applied to a bigger range of scenarios.

## 6.2 Contributions to Artificial Intelligence

The main contributions of this thesis in the context of Artificial Intelligence are the following:

- **A formal monitoring framework supporting operations for expanding and contracting normative contexts**. In §3.3 and §3.4 we have introduced a formal method for monitoring electronic normative environments able to evolve through time as regulations change to adapt to new situations and behaviours. We start by introducing the four operations supported for updating normative environments. This includes operations for expanding and contracting the normative environment, effectively supporting operations to add new norms and remove existing ones (updating an existing norm can be seen as a composite operation consisting in removing a norm and then adding the new version of the norm). Each of these two operations comes in two forms, *Ex Tunc* (*i.e.*, from the outset) and *Ex Nunc* (*i.e.*, from now on) that will retroactively change the normative consequences of actions committed prior to the existence of the norm (from the outset) , or affect only actions committed after the existence of the norm respectively (from now on). In §3.3 we present a summary of a formal base framework (full details are in §A) that we extend to support the four operations as well as institutional powers, including both normative powers and constitutive powers. Once the operations to be supported have been formally defined, the chapter proposes in §3.4 a formal extension of the base normative monitoring framework that will allow it to support normative context modifications at run-time. That is, the normative environment can be expanded and contracted without having to stop the monitoring process. Furthermore, all expansions and contractions performed leave the normative context in a consistent state (*e.g.*, if a norm was violated, it will remain violated, unless an *Ex Tunc* contraction of the norm has been performed). On the one hand the chapter includes informal descriptions of the actions to be performed on the normative context for each of the four supported operations. On the other hand, the chapter also formalizes the impact of the operations in the normative context.

- **Methods for performing the expanding and contracting operations while the monitoring system is on-line**. In §3.5 we operationalise the formal proposal, providing an implementation in the form of a prototype for NoMoDEI. §3 goes on by providing formal algorithms in meta-code. The algorithms implement the four operations formally defined in §3.4.
- **An architectural definition and a working prototype of our framework**. In §3.6 we present the architectural design of the system to be implemented in order to support dynamic normative contexts following our proposed framework. We have partially deployed a prototype §3.7 implementing this architecture , which will be fully documented in the final version of this document. The documentation of the prototype for our formal framework will also include the particular technological stack selected for instantiating the monitoring architecture.
- **Two real world use cases where our framework is applied**. In this document we apply our framework to demonstration scenarios in §4 and §5. As this task is explicitly different from formalizing and deploying the framework we analyse it and draw conclusions *w.r.t.* the application of our framework to the demonstration scenarios in the next section.

## 6.3 Contributions to the Application scenarios

In our proposal we are not only providing a framework for supporting dynamic normative contexts, as seen in the previous section. We are also applying it to demonstration scenarios in order to show the utility and applicability of our framework to real world problems. Specially problems that can be tackled using information technologies in general and multi-agent systems in particular.

We have considered several scenarios where actors (both human and computational) interact in a common environment. We have considered mainly scenarios based on urban mobility, environmental management, interactions in MMORPGs (massive multiplayer online role-playing games) and e-health. Finally we have decided to apply our framework both to environmental management and e-health.

Environmental management provides a test bed where several actors (industries, water treatment plants, towns, *etc.*) interact in a river basin environment. Actors' objectives are heterogeneous and sometimes conflicting. By applying a Normative System we align agent's objectives with common organisational objectives. At the same time, we detect undesirable patterns of behaviour in the agents, such as free raiders, and sanction them. When applying our framework to environmental management we put our focus on how the Normative System can evolve. On the one hand adapting to new regulations and protocols caused by technological advances. On the other hand adapting to unexpected situations which are typically out of control of managers and legislators, such as heavy rains.

E-health provides a test bed where several actors (patients, doctors, relatives *etc.*) interact in a highly regulated medical environment. Actors' interactions must be observed, and medical protocols and regulations enforced. By applying a Normative System we can reflect human regulations and protocols on the information system developed, effectively making it norm and social aware. When applying our framework to a medical scenario we focus on how a dynamic Normative System can provide security, abstraction, flexibility and detect potentially dangerous situations, effectively providing a response in a timely manner.

This section presents conclusions *w.r.t.* the application of our framework to the demonstration scenarios selected, showing how they can benefit from normative frameworks in general and dynamic normative frameworks in particular.

### 6.3.1 Normative Systems for Environmental Management

In §4 we present a norm-aware agent-based model for integrated wastewater management systems. The *Chapter* provides an example on how Normative Systems can be integrated in multi-agent systems where actors' objectives are heterogeneous and sometimes conflicting. The Normative System allows to align agent's objectives with common organisational objectives. At the same time, it allows to detect undesirable patterns of behaviour in the agents, such as free raiders. Thanks to our proposal, misbehaving actors can be sanctioned, effectively enforcing good practices among the actors.

In this aspect, our proposal shows many features in common with several works in the state of the art. However, our proposal goes beyond, as it allows the set of norms governing the multi-agent system to evolve through time. We provide a wide range of examples, where deontic norms in the form of obligations, prohibitions and permissions are inserted, removed and updated. Furthermore, we also show examples of dynamic operations on constitutive norms and constitutive powers.

While most of the systems analysed show a less expressive normative language (they typically do not account for constitutive norms and constitutive powers) we provide a rich set of normative elements, supporting deontic elements (obligations, prohibitions and permissions), constitutive norms, constitutive powers and violation handling norms (*i.e.* sanctions). Furthermore, our normative elements contain a rich structure with activation, maintenance and deactivation conditions, as well as deadlines.

Finally, we support norm dynamics, which is not supported by the proposals analysed in the state of the art. We propose four operations to update the Normative System accounting for norm promulgation and derogation both in prospective and retroactive forms. On the one hand, we combine norm operations with a rich set of normative elements providing a dynamic normative language that can be adapted to a numerous set of contexts and situations. This is specially important in wastewater management scenarios, where the set of norms will evolve adapting to situations which are typically out of control of managers and legislators (*e.g.*, heavy rains, droughts, pollution of the environment). On the other hand we can adapt norms while our system is on-line, inferring a normative state consistent with the update. In scenarios like wastewater management we can not afford to stop monitoring the actions of the different actors, as free raiders and other misbehaving actors could take advantage of this situation.

In contrast, our proposal does not present complex reasoning processes and decision taking mechanisms for the agents involved in the system. We focus on the Normative System, so we can effectively benefit from more expressive and complex agents the other proposals include.

Summarizing, we consider our proposal is a relevant contribution to the state of the art as it allows Normative Systems applied to wastewater treatment scenarios in novel ways. On the one hand we dynamically adapt to new environmental regulations and protocols. Our proposal is able to adapt regulations to exceptional contexts (*e.g.*, heavy rains, river pollution, *etc.*) and adapt them to agent's behaviour to ensure water quality in the river (*e.g.*, implement more restrictive environmental policies if the agents keep polluting the river with the existing ones). On the other hand this allows to define expressive mechanisms to detect non-compliant behaviour and enforce courses of actions that are aligned with the holistic institutional objectives. Our proposal is able to define a wide range of concepts to define and enforce compliant behaviour, including permissions, prohibitions, obligations, constitutive norms and constitutive powers.

In §4, we have seen a wide range of norms and norm operations, inspired in real world regulations and protocols. The chapter focus on how the Normative System can evolve. On the one hand adapting to new regulations and protocols caused by technological advances. On the other hand adapting to unexpected situations which are typically out of control of managers and legislators, such as heavy rains. We have made an effort to provide an example of every possible norm operation, that is, promulgation and derogation in prospective and retroactive form. Furthermore, we have tried to apply each of the norm operations to each different norm type, effectively providing a complete test bed for our framework.

### 6.3.2 Normative systems and E-health

In §5 we present *AVICENA* an extension of the COAALAS project , a framework for multi-agent systems that combines organisational and normative theories with Ambient Assisted Living (AAL) technologies. The COAALAS project has been extended with a dynamic Normative System grounded on the ALIVE framework that allows us to create a computational model for the regulations and protocols around treatment prescription and treatment adherence.

Our proposal is specially relevant when applied to scenarios that are not only regulated, but also contain dynamic regulations. That is, the normative environment to be taken into account (*i.e.* the medical regulations and protocols to be applied) is not static, but changes over time as new regulations arise and obsolete protocols are replaced by modern ones. Changes in normative environments are typically caused by social changes or technological changes. Some examples of such changes are:

- Patient's treatment is updated: Typically, after a visit with his doctor, the patient might have an updated prescription, depending on his evolution and medical history. Therefore, the patient has a new set of obligations and permissions.

- Medical advances: New studies demonstrate a particular habit is non-healthy, and patient's treatment is updated.
- Technological advances: New technologies (*e.g.,* electronic cigarettes, e-prescriptions, *etc.* ) will typically imply an update on medical regulations and protocols.
- Social changes: Habits that were widely accepted in the past (*e.g.,* self-prescription of medications) are not socially accepted anymore.
- Environmental changes: Agent actions can alter the environment, and therefore change the set of norms regulating the system. For example, a patient living in different countries along the year will be subject to different medical regulations. The transition between the different regulations requires a dynamic Normative System.

In order to support such a highly regulated and dynamic scenario we propose a Normative System and integrate it into the *AVICENA* framework. The main contributions of the proposed system *w.r.t.* related work (as presented in §2.4) is proposing a system that meets the following properties:

- **Flexible**: Actors have the flexibility to violate norms (*i.e.* regulations and protocols). When such norms are violated both sanctions (discouraging the agent from following non-complying behaviours in the future) and repair actions (aimed at bringing the system back from an undesirable state to a normal state) are applied. On the one hand it allows us to easily integrate both human and computational actors, as there is no feasible procedure to fully prevent human actors from violating a norm. On the other hand, the level of the sanction applied when violations are detected allows us to define a wide range of norm compliance modes such as:
  - Norms that should never be violated. For instance, it is forbidden to self-prescribe medications. Violating such norms will typically imply a very hard sanction and a repair action.
  - Norms that can be violated in very exceptional cases. For instance, a doctor can only access his patient's medical records, but in case patient's life is at risk and no other doctor is available, a doctor might access the medical records of a patient not assigned to him.
  - Norms that can be violated occasionally: For instance, a patient that is on diet and skips it once a month might remain unsanctioned.
  - Norms that should be violated: By applying positive reinforcement as sanctions, we can effectively detect healthy patterns of behaviour and promote them.
- **Expressive**: We support a wide range of normative concepts. First, deontic norms, including obligations, permissions and prohibitions. Second, constitutive norms, allowing to interpret brute facts from an institutional point of view. This allows abstracting deontic norms effectively providing simple and clear normative contexts. Finally, constitutive powers, allowing to provide institutional meaning to actor actions performed under particular circumstances.
- **Dynamic**: We support norm dynamics, effectively allowing to insert, remove or update norms in the normative context. The normative context can be updated at run-time, while the system is checking if regulations and protocols are abided, and inferring a normative state [UBSA10] consistent with the updates performed (*e.g.,* removing an obligation can remove the sanctions associated to previous violations of the obligation).

We consider our proposal is a relevant contribution to the state of the art. The medical domain is subject to tight regulations and protocols. Therefore the information systems deployed on the medical domain should be subject to such regulations. However, in order to be able to react to unexpected and potentially dangerous situations, informations systems must be flexible, accounting for violations on the regulations and protocols. For instance, a doctor should be able to access a restricted medical record of a patient (at least, partially) if the patient's life is at risk and no other doctor is available. Furthermore, medical regulations and protocols evolve through time, and the information system should be able to reflect this evolution. Therefore, our proposal is relevant, as it is flexible and it allows Normative Systems applied to e-health scenarios to adapt to new regulations and protocols caused by social, technological, medical and contextual changes.

In §5 we have seen a more specific set of norms and protocols, when compared to §4. As §4 already covers every possible norm operation applied to every possible norm type, we have focused in exploring novel applications of norm dynamics in §5. The chapter has three main focus. First, how constitutive norm dynamics add an abstraction layer to the Normative System effectively providing a clear system, easy to maintain and update. For instance, updating medical treatments after visiting a doctor. Second, how constitutive powers add security to the system and can be effectively applied in multi-institutional settings. For instance, where doctors obtain and loose constitutive power as the patient travels between different countries with different normative contexts. Finally, we demonstrate how retroactive promulgation of norms can be used to incentivate healthy behaviours when we use rewards as sanctions. For instance, rewarding a patient with overweight for performing mild physical exercise. The norms presented tackle the following tasks:

- Detect and solve in a timely manner potentially dangerous situations, such as a patient not taking his medication.
- Control security issues related to medical data, such as an unauthorized doctor accessing a patient medical record.
- Control medical protocols, such as preventing self-medication.

Finally, the scenario in §5 has allowed us to test the expressiveness and flexibility of our framework. We have demonstrated the flexibility of sanctions by applying sanctions as rewards to patients following healthy habits. Furthermore, we have demonstrated the utility of retroactive promulgation of norms, even though the operation is forbidden by constitutions and bills of rights in many countries.

## 6.4 Future lines of research

Having support for dynamic normative contexts opens several interesting lines of research, mainly:

- Adaptable normative contexts, able to autonomously insert or remove norms into the system.
- Support for multi-context monitoring, where the monitoring process can jump from one normative context to another just by applying the expansion and contraction operations.
- Scalable networks of monitors deployed on the cloud.

There are another lines of future work, such as extending the Opera model [Dig04] to support our norm modification operations (as outlined in §3.6). However, this section will focus on the first two lines of research, as they are more abstract and open. The third line of research has been explored further during the development of this thesis. Therefore, it contains more results and preliminary work. That is why we consider this line of research deserves its own section, and is introduced in §6.4.3.

### 6.4.1 Adaptable normative contexts

In this PhD thesis we introduce an extended normative framework and architecture to cope with scenarios where the normative context is dynamic, therefore it can expand and contract at runtime. However, we do not account for which updates should be performed on the normative context in order to adapt it to a changing environment and agent population. For instance, we might need to adapt our normative context on the event of unexpected situations, such as traffic jams, heavy rains or a medical emergency during a flight. Furthermore, agent population can also be modified, and a different proportion of free-raiders and miss-behaving agents will account for different norms. More tight norms to account for a higher agent control with more violations and sanctions versus more relaxed norms to minimize monitoring overload.

We have already introduced this concept in §4 and §5, but the actions taken to account for unexpected situations are predefined in the regulations and protocols. Therefore, there is still room for improvement, aiming at having intelligent agents that autonomously decide which norms should be updated and how.

Adapting the normative context to the environment and the behaviour of the agents is an interesting topic, therefore there are several research proposals tackling this issue.

In [MLSRA+15] Morales *et al.*present a proposal for on-line norm synthesis, specially suited for open systems where key aspect of the system are unclear (or even worse, unknown) during design time. In their proposal, they introduce *LION* an algorithm that synthesizes liberal Normative Systems aimed at learning when norms are either substitutable or complementary. The idea is avoiding over-regulation by maintaining only the set of norms that are required to avoid undesirable states of the world. Some systems [MLSRA+13] [MLSRA+14] tackle this issue by providing compact Normative Systems that minimize both the number of norms provided to the agents and the computational effort required to understand them. Other proposals [FT98] try to minimize the number of constraints the Normative System applies to the agents. *LION* aims at:

- Regulate agent interactions in a seamless way.
- Reduce agent's reasoning effort to understand the set of norms in the system by compacting them.
- Impose as few restrictions as possible, to respect agent's autonomy to the greatest extent.

In [VSMS11] Villatoro *et al.*present *Social Instruments* as mechanisms that facilitate the emergence of norms and conventions from the repeated interaction of agents in a society. Norm emergence is a mechanism for sustaining social order, effectively increasing the predictability of the behaviour between the different members of the society and developing the details of unwritten laws. The concept is similar to social learning of norms [MSA08] [SA07] where agents learn norms concurrently by repeating interactions with randomly selected neighbours. Typically we consider a norm as emerged [DPS03] when 90% of the population converges to the same norm. The work in [VSMS11] demonstrates how agents can experience meta-stable subconventions depending on their position in the interaction topology. Subconventions are conventions adopted by a subset of agents in a society who have converged to a different convention than that adopted by the majority of the population. Agents can observe the norm emergence process and rewire the links that relate them to other agents to solve the problem of subconventions.

When applied to self-regulation of agent communities and on-line norm synthesis, the combination of our approach with [MLSRA+15] opens an interesting line of future work for self-adapting Normative Systems. When combined with norm emergence proposals such as [VSMS11] where agents collaborate to create their own norms or norms are created from the observation of agent's interactions the line of future work allows use to simulate and measure the effects of norm emergence in social simulations.

In fact, we have already started exploring this line of research. In [GSANVS13] we contribute to Urban Mobility Decision Support Systems. These systems help policy makers in designing the actions that will improve urban mobility. Typically, this is done via policy optimisation techniques. Current approaches to policy optimisation are based on iterative simulations of a model in order to find an optimal policy combination. Such approaches require fast simulation procedures, as the simulations are performed repetitively with varying parameters. Therefore, they are typically supported by simulations at a macro level, where mathematical formulæ model the behaviour of the system. Simulations at a micro level present higher accuracy but require longer execution times, and therefore they can not be applied to current approaches without resulting in slow optimisation operations.

In [JA] Jiang *et al.* propose a formalization of norm conflicts based on normative states of interrelated norms. Typically, norms are provided by various regulation sources imposed by different institutions. Potential conflicts of values and interests among such institutions may result in inconsistencies among these norms (*e.g.,* a norm obliges a particular action and another norm forbids it). Intuitively, a conflict occurs between an obligation and a prohibition when they constraint on the same behaviour and have an overlapped activation period. That is, both norms are potentially active at the same time (standing for weak conflict) or we can prove they will be active at the same time (standing for hard conflict).

Jiang *et al.* model an institution as a set of norms used to regulate the behaviour of the participating agents, and formalize it using Norm Nets [JADT14]. The formalization includes operational semantics of the norms, with *instantiated*, *activated*, *violated* and *satisfied* states. It is important to note such formalization resembles the norm life cycle we have introduced in this document. The operational semantics are operationalized to Coloured Petri Nets (CPN) [Jen13].

The method for detecting norm conflicts presented by Jiang *et al.* can be effectively attached to our monitoring architecture, analysing norms before introducing them into the system and effectively making sure norm conflicts are avoided. Furthermore, the method is generic and not bound to a specific computational mechanism, but can also be combined with additional conflict detection techniques, such as [LBDV+13]. We consider using Jiang's *et al.* proposal would contribute to improve NoMoDEI, tackling one of our open issues and making it a safer monitoring framework.

In [GSANVS13] we present an approach for policy optimisation based on micro-level simulations where an heuristic provided by the policy maker guides the selection of the scenarios to be simulated. This allows to set a limit on a potentially big search space, and allows for a more accurate selection of the simulation scenarios. The result of the simulations will reveal which is the optimal set of policies to be applied according to policy maker's objectives (ranging from a safe pedestrian-friendly city to an efficient network of streets that maximize vehicle speed). This optimal set of policies can be put in contrast with the actual set of policies in the Normative System, effectively updating it at execution time, and adapting the normative context to the environment. For instance, we can have policies aimed at providing a safe pedestrian-friendly city, but in case of a traffic jam switch to policies aimed at maximizing efficiency in the road network in order to dissolve the traffic jam as soon as possible.

This contributes to the line of research on simulations and Normative Systems. The systems will be able to measure their performance and expand or contract the normative context consequently (*e.g.*, detecting a particular norm is counter-productive for the overall system objectives and remove it from the normative context). We can simulate and analyse the transition period between two Normative Systems, effectively analysing how it does affect system performance, including how long does it take to adopt the Normative System, the performance of the system during the transition (while a particular norm is added or removed) and the consequences to the actors (how the norm is added or removed in terms of retroactive or prospective consequences of the change).

### 6.4.2 Multi-context monitoring

Even though institutions have been studied in the Multi-Agent System community for quite some time, most of the approaches typically focus on specifying a single institution. The issue of modelling multiple interacting institutions has seldomly been addressed. However, institutions are not necessarily separate entities. Several of them could operate within the same context, for instance, governmental legislations in Catalonia (Spain) are typically subject to regional (*i.e.* Catalan), national (*i.e.* Spanish) and European institutions. Not to mention the existence of some international regulations and protocols related to environmental domains (*e.g.,* Kyoto protocol for $CO_2$ emissions) and medical domains (*e.g.,* world health organisation recommendations). Furthermore, agents can participate in several institutions at the same time and move from one institution to another. Finally institutions themselves can be governed by institutions. Therefore, there is an open line of research *w.r.t.* monitoring systems applied to multiple interacting institutions. Some existing lines of research study multi-context institutions.

In [CDVP07b] Cliffe *et al.* propose both a model and operational semantics for specifying individual and collective institutions. They also outline a declarative action language for describing institutions. The idea is modelling particular aspects of a society individually and then combining them to give a richer model. This opens the possibility of using institutions as a means for abstraction and also as a means for delegation. They extend Instql [HCDVP10] [CDVP07a] a formal specification of single institutions to support multi-institutions.

In [CO09] Lopes Cardoso *et al.* propose the idea of agent-based Electronic Institutions instantiated in the scope of a contextual normative background. We can effectively apply our proposal to enrich this interesting line of research. The idea in this case is assigning a monitor instance to every agent in the system. The monitor will interpret the normative state for the agent, notifying norm activation, fulfilment and violation, effectively making the agent norm-aware. Thanks to our dynamic normative framework, the monitor will be able to transition between different normative contexts. The idea of transition is inspired by [CDVP07b]. When peforming the transition, the system can update the set of norms to be taken into account seamlessly. The transition is made on the fly, as we keep monitoring the state of the normative environment for the agent, otherwise we might miss the observation of some important events. Furthermore, the transition is able to cope with multiple normative contexts. That is, if the agent violates a norm in context $C_i$ and then transitions to context $C_j$ the monitor (and by extension, the agent) will be aware norm violation still holds in $C_i$.

### 6.4.3 Improving performance and scalability in normative monitoring

Keeping track of the normative state in a context can be a cumbersome task. Some works such as [ÁNAVSD10] propose a reduction from deontic norms to general production systems. A representation based on production systems is easiest to use at run-time. This allows building a norm monitoring mechanism that can be used both by agents to perceive the actual normative state of the environment and by institutions to detect norm violations and enforce sanctions. The production system can be configured at run-time by adding both abstract organisational specifications (*i.e.*, regulative norms) and sets of counts-as rules (*i.e.*, constitutive norms). In such works, the detection of normative states is a passive procedure that consists in monitoring past events and checking them against a set of active norms.

A clear advantage of a norm monitoring mechanism implemented using general production systems is that the efficiency of the system is bound to the complexity of the system. That is, linear to the number of productions contained in the rules in the worst case and constant in the best case. However, in real world scenarios this might not be enough, as bottlenecks can arise on:

- The number of events received.
- The number of norms (and therefore, the number of rules) loaded in the production system.

As performance is an important aspect on monitoring systems, we have already initiated some preliminary work in the area. In [GSANVS11] we present a prototype based on network analysis of norm dependencies to distribute a norm monitoring process among a set of different interconnected monitors. Our proposal consists in:

- Extending a base formal model to detect information dependencies among the different elements present in the original formal model, starting from low level elements (*i.e.* clause), and extending the formalisation to the higher level elements, including norms and monitors.
- Once we have a method to detect information dependencies we apply it to model the normative context as a graph, where the edges between nodes model information dependencies. Every formula is represented by a node in the graph, and we add an edge between two nodes whenever there is information dependency between the formulas they represent. This has effectively built an structure that supports grouping the most connected formulas together in sub-graphs. In order to detect which are the most connected formulas, we run a *Strongly Connected Components* [NSS94] analysis on the graph modelling the normative framework. This procedure will detect $N$ *Strongly Connected Components*, identifying the formulas belonging to each of them. This process has effectively detected $N$ sub-normative environments containing formulas with high information dependency among them. The sub-normative environments will be connected to each other, as typically there will be information dependencies among the different sub-normative environments. However, the *Strongly Connected Components* analysis guarantees that:
  - The formulas with a high number of information dependencies among them are grouped on the same *Strongly Connected Component*, therefore, they will be on the same sub-normative environment.

  - Information dependencies among formulas in different sub-normative environments are minimized.
- Once we have detected the *Strongly Connected Components* in the graph, we assign one autonomous monitor to each of them. The monitors are running on different computational resources on a cloud, effectively allowing for a distributed computation of the different sub-normative environments. As we need a holistic view of the normative environment, the different monitors are connected to each other via subscription protocols. However, as mentioned before, the *Strongly Connected Components* analysis guarantees that these subscriptions are minimized. An algorithm for performing the normative environment distribution is provided, along with a formula that can be used for deciding when to distribute the normative environment.

Our approach in [GSANVS11] was mainly inspired by the model for distributed interpretation proposed by Lesser and Erman [LE80] In short, the process proposed by Lesser consists on:

- Distributing the problem in several computational resources and give a local view of the problem to each of them.
- Make the computational resources coordinate in order to solve inconsistencies derived from local views of the problem.
- Aggregate the partial solutions provided by each computational node in a global overall solution.

This technique can be effectively applied for distributing the computation of the normative state, effectively allowing to apply the monitoring process to scenarios with a big number of events per unit of time, or a big number of norms to be taken into account. However, the range of problems this technique can be applied to is limited. The problem has to support division into sub-problems, so that distribution and local solving can be performed.

In [UBSA10] Visara Urovi, Stefano Bromuri, Kostas Stathis and Alexander Artikis present a framework for distributing the computation of the social environment. It provides a proof of concept on how distributing the computational process in different computational resources (*e.g.*, in a cloud) allows for larger-scale systems. The framework described provides a simple, yet very clear, example of the idea. And, in our opinion, the main problem of the approach relies in its simplicity. The distribution method introduced is too dependant on being able to model the world as a grid (in order to be balanced, as a rectangular grid). Therefore it can not be applied to scenarios where a physical model of the world is hard or even impossible to assess (*e.g.*, monitoring an auction process, or the messages arriving to an Internet forum). Thus, we consider it a good example, but we would like to provide more generic computational distribution methods, able to be applied to a wider range of environments. Our proposal in [GSANVS11] is similar to Artikis' in the sense that we aim to split the events into distributed and interconnected monitors. However, there is a fundamental difference. Our approach does not rely on a physical division of the normative context, and this provides an important benefit: our approach can be applied to scenarios where the representation of the world is either unclear, hard or even impossible to define (*e.g.,* an auction or the medical domain presented in §5).

In [Che93] Albert Mo King Cheng explores parallel execution as an approach to achieve higher execution speeds in rule-based systems. This work provides an interesting idea about how to model information dependencies among rules as graphs, and how to group highly coupled rules (with high information dependencies among them) using a *Strongly Connected Components* analysis. The way in which information dependencies are detected can be improved and made more expressive. Applying the procedure to a formal framework rather than to a programming language source code would make it more generic, able to be applied in a wider range of scenarios. Finally, nowadays it seems to make more sense to apply the procedure to cloud computing techniques rather than to parallel computing ones.

The framework for parallel execution of rules in production systems proposed by Anoop Gupta *et al.* in [GFK$^+$88] provides an interesting application of the divide and conquer paradigm. A complex problem is divided into N local sub-problems plus a coordination problem. The local sub-problems are executed in parallel and the coordination problem puts the local results together resolving possible information conflicts. In general, the process results in an overall performance gains. As long as we are careful with the overheads. Again, we consider the idea to be interesting and applicable to our framework for normative computation distribution. However, we consider applying the procedure to a formal framework rather than to a specific implementation of a production rules engine would make it more generic, able to be applied in a wider range of scenarios. Again, it seems better to apply the procedure to computing techniques rather than to parallel computing ones.

When compared to Cheng's and Gupta's work, our approach in [GSANVS11] is more generic, as we are building a dependency graph of a set of norms, no matter how these norms are computed internally by each monitor (either production rules, Java programs or alternative approaches). Furthermore, our approach has been adapted to new techniques, mainly, substituting parallelism for distributed computing.

As future work we plan to apply our proposal in [GSANVS11] to the framework introduced in this thesis, effectively extending it with parallel computing capabilities. The extension will provide support for real time monitoring even on scenarios where the number of norms loaded in the production system is high, as the monitoring framework can distribute the normative context among a set of collaborative monitoring systems at runtime. The main goal is to effectively reduce the number of events each monitor has to take into account, allowing each of them to compute a smaller part of the whole normative context.

As a holistic view of the whole monitoring state is required, monitors will be linked together in order to be able to collectively infer the full normative state of the context. However, if the communication overhead required for this process is too high, it might overpass the benefits of distributing the normative context computation. This fact can lead to scenarios where the performance of the distributing normative context computation is lower than the performance of a centralized normative context computation. In order to avoid this issue, the model presented in this proposal guarantees links between distributed monitors are kept to the minimum in order to minimise dependencies.

In order to support the creation and maintenance of the network of monitors, our proposal for dynamic normative contexts comes in handy. It will allow to create a network of monitors at runtime, and update them (*e.g.,* for load balancing) adapting them to the flow of events to be monitored. Norms can be added and removed from the different monitors on the fly, without having to stop them, effectively allowing to distribute the work load between the network of collaborating monitors at run time. Thanks to our proposal the monitors can split and merge dynamically, effectively adapting themselves to the monitoring load of the system. They can even delegate the monitoring of a set of norms to another monitor in order to deal with peaks in the number of events to be observed.

APPENDIX A

# Analysis of relevant monitoring frameworks

This annex contains an in-depth analysis of the lines of work for norm monitoring analysed in *Section* 2.1.6 and *Section* 2.1.7. Such an extensive analysis is performed because we have used these formal frameworks as basis for the NoMoDEI framework presented in §3. Therefore, this section includes a complete analysis the reader can refer to in case further details on these works are needed.

## A.1 Towards a Formalisation of Electronic Contracting Environments

This section contains a formal approach to Oren *et al.* work. We consider it to be a good detailed analysis to the concepts they define in their work. They formalise their notion of norms for modelling contract clauses via the following steps:

- Formal preliminaries. Introducing the basic concepts required to understand the rest of the formalisation steps.
- Definition of the structure of a contract in general, and the structure of a contract clause (*i.e.*, norm) in particular
- Definition of the operational semantics, defining how the status of a norm can change over time.

Therefore, this section will follow the same approach.

### A.1.1 Formal Preliminaries

In their work, they use a predicate based first order language $\mathcal{L}$ where the following elements are available:

- Connectives: $\{\neg, \wedge, \vee, \rightarrow\}$
- Infinite set of variables.
- Non-logical predicate symbol.
- Constant symbol.
- Function symbol.

The set of all well-formed formulae of $\mathcal{L}$ is denoted as $wff(\mathcal{L})$ and a single well-formed formula as $wff$. $\mathcal{S} = \langle t1/v1, ..., tn/vn\rangle$ is a substitution of terms $t$ for variables $v$ in a given $wff$. If no variables exists in a given $wff$ after a substitution, it is said to be a fully grounded $wff$. Otherwise, it is a partially grounded $wff$.

In Oren *et al.* work, the truth values of the different predicates can be inferred from sources such as ontologies, an action model of the environment or another sources. Each of these different sources, generates a theory denoted by $\Gamma$. Theories, can be labelled depending on the source generating them. A typical theory is the one generated by the environment source, labelled as $\Gamma_{Env}$. The union of all the theories is denoted by a label-less theory $\Gamma$. Finally, a contract document contains a set of clauses modelling the different norms imposed on agents. Since a contract document can be instantiated several times with different sets of agents associated to similar roles, they identify the agents on a Contract via an indirection mechanism. That is, a contract document imposes norms on a set of roles, and agents are associated with these roles.

### A.1.2 Structural definitions

Once the formal preliminaries assumed by Oren *et al.* in their work have been introduced, we introduce their definition of the structure of norms and contract documents. The structure contains the following elements: Agent names and roles, norms, norm instantiations and contracts.

**Agent Names and Roles**

Agents in Oren *et al.* framework are left unspecified. They can be implemented in any *Agent Programming Language* and hosted in any *Agent Platform*. Only one assumption is made: Agents are associated with a unique agent name. Therefore, this name will be used to associate agents to specific norms. Contracts, associate agents with roles. For instance, a contract specifying a teacher must be available at the University at least 2 hours per week for solving student's doubts will contain two roles. *Teacher* and *Student*. A specific agent *Isangi* may then be associated with the *Teacher* role in a particular instantiation of the contract. If the agent *Isangi* happens to be a student too (*e.g.*, in a case of PhD student who does some lecturing) it could be associated with the *Student* role in a different instantiation of the same contract. A hierarchical relation is defined on contract roles. Whenever a particular agent is assigned to a role is also assigned to it's parents roles. Therefore, when a given agent assumes the clauses applying to a particular role, it also assumes the clauses applying to the parent's roles.

**Definition A 1** ***(Role)***

*A role is a constant. The set of all roles is denoted by Roles. The hierarchical relations among roles are captured by the RoleHierarchy element, which is a binary relation of the form (ParentRole, ChildRole) where* $ParentRole, ChildRole \in Roles$. □

**Norms**

As stated before, a Contract in Oren *et al.* framework is a set of clauses represented by norms. A norm consists in the following components.

- Type. States weather the norm is an obligation or a permission.
- Activation condition. States when the norm must be instantiated.
- Normative state. Identifies when the norm is violated (in the case of obligations) or what the agents are allowed to do (in the case of permissions or prohibitions).
- Expiration condition. States when the norm no longer affects the agent.

- Target. Set of agents affected by the norm.

Norms pass from one state to another (*i.e.*, activated, met or discharged) based on factors such as the environment and the status of the contracts (the status of the norms modelling the clauses in the contract to be more specific). They define a theory as introduced in their *Formal Preliminaries* known as *Normative Environment Theory* $\Gamma_{NEnv}$. Needless to say, $\Gamma_{NEnv} \in \Gamma$. The *Normative Environment Theory* allows for effectively interpreting the status of norms. Oren *et al.* represent norms as a norm type and 4 well-formed formulas:

**Definition A 2** ***(Abstract Norm)***
*A norm is a tuple of the form*

$$\langle NormType, NormActivation, NormCondition, NormExpiration, NormTarget\rangle$$

*Where:*
$NormType \in \{obligation, permission\}$
$NormActivation \in wff \wedge NormCondition \in wff \wedge NormExpiration \in wff \wedge NormTarget \in wff$ □

An *Abstract Norm Store* is a set of abstract norms, and is denoted as $\mathcal{ANS}$.

The set of substitution variables $S$ that make the Activation Condition become true is applied to the other components of the abstract norm, effectively resulting in a instantiated norm. They do not specify what happens in the case multiple instantiations from a particular abstract norm are possible (*i.e.*, multiple substitutions make the activation condition become true). This is specially relevant in cases of norms that are always active, therefore, having an activation condition set as *True*.

**Norm Instantiation**

A norm that can be applied in a wide range of situations is considered to be abstract. When one of the situations in which the norm can apply is met, the norm is instantiated. Then and only then, the norm exerts a normative force on the agents associated to it.

From a structural point of view , the only difference between an instantiated norm and an abstract norm, is that in an instantiated norm, the activation condition is fully grounded, and its remaining parameters are either fully or partially grounded using the same grounding as in the activation condition.

**Definition A 3** ***(Instantiated Norm)***
*Given an abstract norm of the form:*

$$\langle NormType, NormActivation, NormCondition, NormExpiration, NormTarget\rangle$$

*It can be instantiated using a set of theories $\Gamma$ including, at least, one theory representing the domain environment $\Gamma_{Env}$ and one representing the Normative Environment $\Gamma_{NEnv}$. The instantiation process results on a instantiated norm of the form:*

$$\langle NormType, NormActivation', NormCondition', NormExpiration', NormTarget'\rangle$$

*Where the following properties are met:*

- $\Gamma \vdash NormActivation'$
- *NormActivation′ is fully grounded*
- $NormActivation' = S(NormActivation)$

- $NormCondition' = S(NormCondition)$
- $NormExpiration' = S(NormExpiration)$
- $NormTarget' \in AgentNames$
- $NormTarget' = \langle X | \Gamma \cup \langle NormActivation' \cup \langle S(NormTarget) \rangle \vdash X \rangle$

□

As stated before, *NormTarget'* is the set of individuals (*i.e.*, agents) to whom the instantiated norm applies. As seen in the previous definition, this set of individuals is entailed via both the domain environment theory (*i.e.*, $\Gamma_{Env}$) and the normative environment theory (*i.e.*, $\Gamma_{NEnv}$). The normative environment theory may be used when a target has to be identified based on the status of another norm. Typically, it happens in the case of *contrary to duty obligations* where a penalty must be paid by the agents violating a given norm.

**Contract**

Once the structural definition of Roles and Norms has been provided, they are able to define the structure of Contracts. Formally:

**Definition A 4** ***(Contract)***

*A contract (Contract Document) is a tuple with the following form:*

$$\langle \Gamma, Norms, Roles, RoleMapping \rangle$$

*Where:*

- $\Gamma$ *is the set of all theories*
- *Norms is a set of abstract norms*
- *Roles is a set of role definitions*
- *RoleMapping maps role definitions in the Roles set to the set of Agent Names which form the Contract Parties.*

*The set ContractParties $\subseteq$ AgentNames is the set of agents named within the contract.*
*It implies the following condition must hold:*

$$\Gamma \cup Roles \cup RoleMapping \vdash X \textit{ where } X \subseteq ContractParties$$

*This condition raises a requirement. A contract document might only impose norms on contract parties.* □

Oren *et al.* allow for its main elements, and specially contracts, to have additional meta-data associated. Typically, a piece of meta-data associated with contracts is the contract status. They vies metadata as an additional theory from which agents can infer information, $\Gamma_{metadata}$.

### A.1.3 Operational Semantics

Once the structure of the different elements has been introduced, they introduce the operational semantics for these elements. This allows for defining how, given a particular evolution of the environment, norms are instantiated fulfilled, violated and discharged. For this, they introduce the Normative Environment theory $\Gamma_{Env}$ which defines predicates that are useful for identifying the status of norms as they progress through their lifecycle.

**Definition A 5** ***(Normative Environment Theory)***
*A Normative Enviroment Theory is a set of Normative States:*

$$\Gamma_{NEnv} = \langle NS_1, NS_2 \rangle$$

*Where each normative state keeps track of four basic events:*

- *An abstract norm is instantiated*
- *An instantiated norm expires*
- *A normative condition inside a norm, holds*
- *A normative condition inside a norm, does not hold*

□

Therefore, in order to define a normative state, predicates based on Norm's *NormCondition* and *ExpirationCondition* elements must be defined. Formally:

**Definition A 6** ***(Predicates)***
*Let **in** be an instantiated norm of the following form:*
$in = \langle NormType, NormActivation, NormCondition, NormExpiration, NormTarget \rangle$
*Then, for* $N \in \{NormCondition, NormExpiration\}$ *predicate* $holds(in, N)$ *evaluates to:*
- *True if*
  - $\Gamma \vdash N'$
  - $N'$ *is entailed with all variables in* $N$ *grounded.*
- *Otherwise*

□

Oren *et al.* definition of a normative state identifies the instantiated norms whose normative condition evaluates to true, the ones whose normative condition evaluates to false, and the ones whose expiration condition evaluates to true. Formally:

**Definition A 7** ***(Normative State)***
*Let INS be a set of instantiated norms,* $\Gamma_{Env}$ *a domain environment theory and* $\Gamma NEnv$ *a Normative environment theory. A Normative State NS is a tuple of the form:* $\langle NSTrue, NSFalse, NSExpires \rangle$ *where:*
- *NSTrue* = $\{in \in INS | holds(in, NormCondition) == true\}$
- *NSFalse* = $\{in \in INS | holds(in, NormCondition) == false\}$
- *NSExpires* = $\{in \in INS | holds(in, NormExpiration) == true\}$

□

It must be noted that $NSExpires \subseteq NSTrue \cup NSFalse$ holds. Therefore, in order to identify the set of instantiated norms in a particular normative state, denoted by $inst\_norms(NS)$, it is enough to identify the set of norms whose normative condition evaluates to true and the set of norms whose normative condition evaluates to false. Formally:

**Definition A 8** ***(Instantiated Norms)***
*Given a Normative State NS:*

$$inst\_norms(NS) = NSTrue \cup NSFalse$$

□

They define a *Normative Environment* as an ordered sequence of normative states $\langle NS_1, NS_2...NSi\rangle$ where the normative state $NS_i$ is prior to the normative state $NS_{i+1}$. For any normative state, they define the next normative state by removing the expired instantiated norms, adding new instantiated norms and checking the norm state of all instantiated norms. Therefore, in order for normative state advance to be efficient, these three operations must be performed with the lowest computational cost possible. They define normative state semantics to formally specify the transitions from a normative state to the next one:

**Definition A 9** ***(Normative State Semantics)***
*Let ANS be an abstract norm store and NE a normative environment of the following form:*

$$\langle NS_1, NS_2...NSi\rangle$$

*Let $\Gamma_i$ be the set of well-formed-formulas denoting the domain environment associated with $NS_i$.*

*The set of potential norms for $NS_i$ is defined as:*

- *The norms that are instantiated in the previous state $NS_{i-1}(inst\_norms(NSi-1))$*
- *The norms in the abstract norm store that are instantiated with respect to $\Gamma i$ (that is, inst(ANS))*
- *The norms that have not expired in the previous state $NSExpires_{i-1} == false$*

*Formally:*

$$PNorms_i = inst\_norms(NS_{i-1} \cup inst(ANS) \setminus NSExpires_{i-1}$$

*Then, $NS_i = \langle NSTrue_i, NSFalse_i, NSExpires_i\rangle$ can be defined with respect to the set of norms $Pnorms_i$ and the theory $\Gamma_i$.*

*An initial normative state can be defined as follows:*

$$\langle NSTrue_0, NSFalse_0, NSExpires_0\rangle$$

*Where*

- $NSTrue_0 = \{\}$
- $NSFalse_0 = \{\}$
- $NSExpires_0 = \{\}$

□

In order to easily introduce the following definitions presented in Oren *et al.* work we assume a normative environment $\{NS_1, NS_2, ...NS_n\}$ where $NS_i = \langle NSTrue_i, NSFalse_i, NSExpires_i\rangle$ and $1 < i < N$. The *Gödelisation* operator is used for naming normative states, allowing them to be used within well-formed-formulas. For instance $\lceil NS_i \rceil$ names the normative state $NS_i$

The next concept they declare is the *instantiated* predicate. Informally, given a normative state $NS_i$ and a norm $in$ $instantiated(\lceil NS_i \rceil, in)$ evaluates to true if the norm $in$ is instantiated in $NS_i$ and was not instantiated in $NS_{i-1}$ or not expired in $NS_{i-1}$, and therefore, can still be instantiated in $NS_i$. Formally:

**Definition A 10** ***(The instantiated predicate)***

$$\Gamma_{NEnv} \vdash instantiated(\lceil NS_i \rceil, in)$$

*iff*

$$in \in inst\_norms(NS_i) \wedge (in \notin inst\_norms(NS_{i-1}) \vee in \notin NSExpires_{i-1})$$

*It is assumed by default*

$$\Gamma_{NEnv} \nvdash instantiated(\lceil NS_0 \rceil, in)$$

□

Then, the *expires* predicate is defined. It holds true whenever an instantiated norm expired within a specific normative state. Formally:

**Definition A 11** ***(The expires predicate)***
*Given an instantiated norm $in$ and a normative state $\lceil NS_i \rceil$:*

$$\Gamma_{NEnv} \vdash expires(\lceil NS_i \rceil, in)$$

*iff*

$$in \in NSExpires_i$$

*It is assumed by default*

$$\Gamma_{NEnv} \nvdash expires(\lceil NS_0 \rceil, in)$$

□

The *active* predicate holds if a norm is instantiated within a normative state. It can happen for two reasons: Either the norm was instantiated within the state or it was instantiated earlier and has not expired yet. Formally:

**Definition A 12** ***(The active predicate)***
*Given an instantiated norm $in$ and a normative state $\lceil NS_i \rceil$:*

$$\Gamma_{NEnv} \vdash active(\lceil NS_i \rceil, in)$$

*iff*

$$instantiated(\lceil NS_i \rceil, in) \vee (in \in inst\_norms(NS_{i-1}) \wedge in \notin NSExpires_{i-1}))$$

*It is assumed by default*

$$\Gamma_{NEnv} \nvdash active(\lceil NS_0 \rceil, in)$$

□

The *becomesTrue* predicate holds if a norm's normative condition evaluates to true and either the normative condition evaluated false on the previous normative state or the norm has been instantiated in the current normative state. Formally:

**Definition A 13** ***(The becomesTrue predicate)***
*Given an instantiated norm $in$ and a normative state $\lceil NS_i \rceil$:*

$$\Gamma_{NEnv} \vdash becomesTrue(\lceil NS_i \rceil, in)$$

*iff*

$$in \in NSTrue_i \wedge (in \in NSFalse_{i-1}) \vee instantiated(\lceil NS_i \rceil, in))$$

□

The *becomesFalse* predicate is analogous to the *becomesTrue* predicate, dealing with the negation of the normative condition rather than with the normative condition itself. Formally:

**Definition A 14** ***(The becomesFalse predicate)***
*Given an instantiated norm $in$ and a normative state $\lceil NS_i \rceil$:*

$$\Gamma_{NEnv} \vdash becomesFalse(\lceil NS_i \rceil, in)$$

*iff*

$$in \in NSFalse_i \wedge (in \in NSTrue_{i-1}) \vee instantiated(\lceil NS_i \rceil, in))$$

□

They use both *becomesTrue* and *becomesFalse* predicates to define the *isTrue* and *isFalse* predicates. These predicates check weather a norm is active and true (or false) in some normative state. Formally:

**Definition A 15** ***(The isTrue predicate)***
*Given an instantiated norm $in$ and a normative state $\lceil NS_i \rceil$:*

$$\Gamma_{NEnv} \vdash isTrue(\lceil NS_i \rceil, in)$$

*iff*

$$becomesTrue(\lceil NS_i \rceil, in) \vee (active(\lceil NS_i \rceil, in) \wedge in \in NSTrue_{i-1})$$

□

**Definition A 16** ***(The isFalse predicate)***
*Given an instantiated norm $in$ and a normative state $\lceil NS_i \rceil$:*

$$\Gamma_{NEnv} \vdash isFalse(\lceil NS_i \rceil, in)$$

*iff*

$$becomesFalse(\lceil NS_i \rceil, in) \lor (active(\lceil NS_i \rceil, in) \land in \in NSFalse_{i-1})$$

□

In order to fully understand the *isFalse* and *isTrue* predicates definitions, it must be taken into account they are assuming a closed world model. That is, if a given predicate can not be entailed from a Normative environment, then the negation of the predicate can be entailed. Formally:

**Definition A 17** ***(Properties of the normative environments)***
*Given a predicate $x$ and a normative environment $\Gamma_{NEnv}$:*

$$\Gamma_{NEnv} \vdash x$$

*iff*

$$\Gamma_{NEnv} \nvdash \not x$$

*This has two major implications:*
- $\Gamma_{NEnv} \nvdash \perp$
- $\neg$ *is given negation as failure semantics*

□

Once this set of low level predicates has been defined, they use them to define additional predicates. Oren *et al.* start by defining a set of predicates able to operate both on abstract and instantiated norms.

**Definition A 18** ***(Norm access predicates)***
*Given the following elements:*
- *A norm $N$*
- *A norm type $Type$ associated with $N$*
- *A norm activation condition $NormActivation$ associated with $N$*
- *A norm expiration condition $NormExpiration$ associated with $N$*
- *A norm target set $NormTarget$ associated with $N$*
- *The state component of N's normative condition condition $SMainteinanceCondition$*
- *The action component of N's normative condition condition $AMainteinanceCondition$*

*The following predicates are defined:*
- $type(N, X) = true$ *iff* $X = Type$ *and false otherwise*
- $normActivation(N, X) = true$ *iff* $NormActivation$ *unifies to* $X$ *and false otherwise*
- $normSCondition(N, X) = true$ *iff* $SMainteinanceCondition$ *unifies to* $X$ *and false otherwise*
- $normACondition(N, X) = true$ *iff* $AMainteinanceCondition$ *unifies to* $X$ *and false otherwise*
- $normExpiration(N, X) = true$ *iff* $NormExpiration$ *unifies to* $X$ *and false otherwise*
- $normTarget(N, A) = true$ *iff there is a unification between some element of* $A$ *and* $NormTarget$

□

Finally, they go on by defining predicates that are based on the normative environment. They will form the base of Oren *et al.* normative environment theory $\Gamma_{NEnv}$. They define the *violated*, *fulfilled*, *violationHandler* and *handlesViolation* predicates as follows:

**Definition A 19** ***(Normative Environment Predicates)***
*Given an instantiated norm $in$, a normative state $\lceil NS_i \rceil$ and the instantiated norms $N$, $N_1$ and $N_2$:*

$$violated(\lceil NS_i \rceil, in) = normType(in, obligation) \wedge isFalse(\lceil NS_i \rceil, in)$$

$$fulfilled(\lceil NS_i \rceil, in) = expires(\lceil NS_i \rceil, in) \wedge \neg violated(\lceil NS_i \rceil, in)$$

$$fulfilled(\lceil NS_i \rceil, in) = expires(\lceil NS_i \rceil, in) \wedge violated(\lceil NS_i \rceil, in)$$

$$violationHandler(N) = normActivation(N, \lceil violated(\lceil NS_i \rceil, in) \rceil$$

$$handlesViolation(N_1, N_2) = normActivation(N_1, \lceil violated(\lceil NS_i \rceil, N_2) \rceil$$

□

## A.2 Normative Monitoring: Semantics and Implementation

This section presents a detailed analysis of Alvarez-Napagao *et al.* formal approach. It starts by introducing a formalism for both regulative (deontic) and substantive (constitutive) norms based on *Structural Operational Semantics*. Goes on by introducing the reduction from *Structural Operational Semantics* to *Production System* semantics. And the section ends by introducing an implementation compliant to the *Structural Operational Semantics* and the *Production System* semantics.

### A.2.1 Formal Semantics

This subsection outlines the formal semantics of Alvarez-Napagao *et al.* framework. The subsection starts by providing some formal preliminary definitions that will be used on the rest of the formalisations. The subsection goes on by providing the semantics of institutions. The approach is based on the environment specifying the regulative and constitutive norms. The subsection finishes by providing the details on how the institution evolves over time based on events, and the impact on the monitoring process.

In their work, Alvarez-Napagao *et al.* use a predicate based first order language $\mathcal{L}_O$ where the following elements are available:

- Connectives: $\{\neg, \wedge, \vee, \rightarrow\}$
- Infinite set of variables.
- Predicates taken from an Ontology $\mathcal{O}$.
- Constants taken from an Ontology $\mathcal{O}$.
- Function symbol.

The set of all well-formed formulae of $\mathcal{L}_O$ is denoted as $wff(\mathcal{L}_O)$ and a single well-formed formula as $wff$. It assumed well-formed formulae are in *Disjunctive Normal Form* (DNF).

$\Theta = \langle t1 \rightarrow v1, ..., tn \rightarrow vn \rangle$ is a substitution instance of terms $t$ for variables $v$ in a given $wff$. If no variables exists in a given $wff$ after a substitution, it is said to be a fully grounded $wff$. Otherwise, it is a partially grounded $wff$.

The set of roles in a normative system is denoted as a set of constants known as *R*. This set is defined on the Ontology $R \in O$. The set of participants is denoted by *P* where each participant enacts, at least, one role according to the ontology *O*.

A norm is a tuple with the following elements:

- Activation condition: Specifies when a norm becomes active. Is the triggering element in the norm instantiation process. When the different conditions in the norm's activation condition hold, the norm is effectively instantiated, creating a new norm instance. Variables are included in the norm representation, therefore, multiple instantiations of the same norm can be handled and tracked separately.
- Maintenance condition: Specifies a condition that, when no longer hold, leads to the violation of the norm.
- Norm deadline: Representing one or several deadlines for the norm to be fulfilled.
- Target condition: Specifies a condition that, when hold, leads to the fulfilment of the norm.
- Deactivation condition: Defines when the norm becomes inactive. It will typically correspond to the target condition (that is, fulfilling the norm instance deactivates that instance of the norm) but more conditions can be added for representing different deactivation scenarios.
- Norm subject: Set of roles in the ontology the norm applies to.

Formally a norm is:

**Definition B 1** ***(Norm)***
*A norm $n$ is a tuple $n = \langle f_A, f_M, f_\delta, f_D, f_w, w \rangle$ where:*

- *$f_A$ represents the activation condition of the norm. $f_A \in wff(\mathcal{L}_O)$*
- *$f_M$ represents the maintenance condition of the norm. $f_M \in wff(\mathcal{L}_O)$*
- *$f_D$ represents the deactivation condition of the norm. $f_D \in wff(\mathcal{L}_O)$*
- *$f_\delta$ represents the deadline condition of the norm. $f_\delta \in wff(\mathcal{L}_O)$*
- *$f_w$ represents the target condition of the norm. $f_w \in wff(\mathcal{L}_O)$*
- *$w$ represents the subject of the norm. $w \in R$*

It must be noticed that, although the norm representation presented does not explicitly include deontic operators, the combination of the activation, deactivation and maintenance conditions is as explicit as conditional deontic statements with deadlines. It is even able to express unconditional norms and maintenance obligations. For proving this statement, the following relations must be taken into account:

- $f_\delta \rightarrow \neg f_M$, since the maintenance condition is more expressive than the deadline alone.
- $f_w \rightarrow f_D$, since the deactivation condition specifies the norm is fulfilled, or some special deactivation scenario holds.

Intuitively, after the norm activation condition holds, the norm subject has the obligation to see to it that the norm target holds true, before the norm maintenance condition is negated (that is, either the norm deadline has been reached or some special deactivation scenario comes into place) until the norm deactivation condition holds (that is, the norm is fulfilled or has expired). Formally, using a formalism for conditional deontic statements with deadlines as the one in [DBDM04]:

**Definition B 2** ***(Relation to conditional deontic statements with deadlines)***

$$f_A \rightarrow [O_w(E_w f_w \leq \neg f_M)\boldsymbol{U} f_D]$$

*Where:*

*$E_a p$ stands for agent $a$ sees to it $p$ becomes true*

***U** is the $CTL^*$ until operator*

$N$ denotes a set of norms. A particular set of norms is the set of *violation handling norms*. These norms are activated automatically by the violation of another norm. Violation handling norms are used as *sanctioning norms* if they are to be fulfilled by the norm violating actor (*e.g.*, the obligation of a driver to pay a fine if he/she crashes on a traffic sign) and used as *reparation norm* if they are to be fulfilled by an institutional actor (*e.g.*, the obligation of the traffic authorities to repair the traffic sign broken during the crash). Formally:

**Definition B 3** ***(Violation handling norm)***
*Given two norms $n$ and $n'$ of the form:*

$$n' = \langle f'_A, f'_M, f'_\delta, f'_D, f'_w, w' \rangle$$

$$n = \langle f_A, f_M, f_\delta, f_D, f_w, w \rangle$$

*$n'$ is a violation handing norm of $n$, denoted by $n \rightsquigarrow n'$ iff: $f_A \wedge \neg f_M \wedge \neg f_D \equiv f'_A$*

They define norms in an abstract manner, affecting all possible participants enacting a particular role. When a norm is active there is a norm instance of the form $ni = \langle n, \theta \rangle$ for a given norm $n$ and a particular substitution instance $\theta$.

A state of the world $s_t$ is the set of predicates holding at a specific point of time $t$. The set of predicates is defined on the ontology, $s_t \in O$. $S$ is the set of all possible states of the world, $S \in \mathcal{P}(O)$. An expansion of a state of the world, denoted by $F(s)$, is the minimal subset of elements of $wff(\mathcal{L}_O)$ using the predicates in $s$ in combination with the logical connectives $\{\neg, \wedge, \vee, \rightarrow\}$.

Alvarez-Napagao *et al.* approach fills the need for the interpretation of brute events as institutional facts (also known as *constitution of social reality*) by using *counts-as rules*. Counts-as rules are tuples containing the following elements:

- A brute fact. Brute facts are elements of the set of all well-formed formulae.
- An institutional fact. Institutional facts are elements of the set of all well-formed formulae.
- A context. A set of predicates, that is, a state of the world (or a subset of a state of the world).

Counts-as rules state that, in a given context, a brute fact counts-as an institutional fact. Formally:

**Definition B 4** ***(Counts-as rules)***
*A counts-as rule is a tuple of the form:*

$$c = \langle \gamma 1, \gamma 2, s \rangle$$

*where:*

$$c = \gamma 1, \gamma 2 \in wff(\mathcal{L}_O)$$

$$c = s \subseteq O$$

$C$ *denotes a set of counts-as rules*

Using the above preliminary definitions they are able introduce the concept of Institution. Formally:

**Definition B 5** ***(Institution)***
*A institution is a tuple of the form:*

$$c = \langle N, R, P, C, O \rangle$$

*where:*

$N$ *is a set of norms*

$R$ *is a set of roles*

$P$ *is a set of participants*

$C$ *is a set of counts-as rules*

$O$ *is a set an Ontology*

By defining the concept of institution, they are able introduce the concept of normative monitor. A *Normative Monitor* is composed of an institutional specification. This specification includes, norms, the current state of the world and the current normative state. In order to track the normative state of an institution in a particular point of time, three norm instance sets are defined:

- Instantiation set $IS$.
- Fulfilment set $FS$.
- Violation set $VS$.

Formally:

**Definition B 6** ***(Norm life cycle)***
*Let:*

- $n$ *be a norm of the form* $n = \langle f_A, f_M, f_\delta, f_D, f_w, w \rangle$.
- $n_i$ *be a norm instance of the form form* $ni = \langle n, \theta \rangle$.
- $s$ *be a state of the world with an expansion* $F(s)$.

*The life cycle of the norm instance* $n_i$ *is defined by the following normative state predicates:*

- $activated(n_i) \Leftrightarrow \exists f \in F(s) : \Theta(f_A) \equiv f$
- $maintained(n_i) \Leftrightarrow \exists \Theta' \wedge \exists f \in F(s) : \Theta'(f_M) \equiv f \wedge \Theta' \subseteq \Theta$
- $deactivated(n_i) \Leftrightarrow \exists \Theta' \wedge \exists f \in F(s) : \Theta'(f_D) \equiv f \wedge \Theta' \subseteq \Theta$
- $instantiated(n_i) \Leftrightarrow ni \in IS$
- $violated(n_i) \Leftrightarrow ni \in VS$
- $fulfilled(n_i) \Leftrightarrow ni \in FS$

Then, a normative monitor $M_I$ for an institution $I$ can be defined as a tuple of elements. Formally:

**Definition B 7** ***(Normative Monitor)***
*A Normative Monitor $M_I$ for an institution $I$ is a tuple of the following form:*

$$M_I = \langle I, S, IS, VS, FS \rangle$$

*Where:*

- $I = \langle N, R, P, C, O \rangle$.
- $S = \mathcal{P}(O)$.
- $IS = \mathcal{P}(N \times S \times Dom(S))$.
- $VS = \mathcal{P}(N \times S \times Dom(S))$.
- $FS = \mathcal{P}(N \times S \times Dom(S))$.

$\Gamma$ denotes the set of all possible configurations of a Normative Monitor $M_I$. $\Gamma = I \times S \times IS \times VS \times FS$.

However, the definition above does not take into account the dynamic aspect of incoming events affecting the state of the world through time. Therefore, Alvarez-Napagao *et al.* extend their model assuming there a continuous sequential stream of events received by the monitor. Formally:

**Definition B 8** ***(Event)***
*An event is a tuple of the following form:*

$$e = \langle \alpha, p \rangle$$

*Where:*

- $\alpha \in P^3$.
- $p \in S$ *and is fully grounded.*

$E$ denotes the set of all possible events, $E = \mathcal{P}(N \times S)$.

Finally, Alvarez-Napagao *et al.* define the concept of *Labelled Transition System* for a *Normative Monitor* $M_I$. Formally:

**Definition B 9** ***(Labelled Transition System)***
*The Labelled Transition System for a Normative Monitor is defined by $\langle \Gamma, E, \triangleright \rangle$.*
*Where:*

- $E$ *is the set of all possible events* $e = \langle \alpha, p \rangle$.
- $\triangleright$ *is a transition relation such that* $\triangleright \subseteq \Gamma \times E \times \Gamma$.

Please, notice the inference rules for the transition relation $\triangleright$ are depicted in *Figure*A.1.

Event processed:

$$\frac{e_i = \langle \alpha, t, p\rangle}{\langle s\rangle \overset{ep}{\rhd} \langle s \cup \{p\}\rangle}$$

Counts-as rule activation:

$$\frac{\begin{array}{c}\Theta(\gamma_1) \equiv f \qquad \Theta(\gamma_2) \notin s \\ \langle \gamma_1, \gamma_2, s_i\rangle \in C \qquad f \in F(s) \qquad s_i \subseteq s\end{array}}{\langle s\rangle \rhd \langle s \cup \{\Theta(\gamma_2)\}\rangle}$$

Counts-as rule deactivation:

$$\frac{\begin{array}{c}\Theta(\gamma_1) \equiv f \qquad \Theta(\gamma_2) \in s \\ \langle \gamma_1, \gamma_2, s_i\rangle \in C \qquad f \in F(s) \qquad s_i \not\subseteq s\end{array}}{\langle s\rangle \rhd \langle s - \{\Theta(\gamma_2)\}\rangle}$$

Norm instantiation:

$$\frac{\begin{array}{c}activated(n, \Theta) \\ n \in N \qquad \neg\exists n' \in N, n' \rightsquigarrow n \qquad \langle n, \Theta\rangle \notin is\end{array}}{\langle is\rangle \overset{nii}{\rhd} \langle is \cup \{\langle n, \Theta\rangle\}\rangle}$$

Norm instance violation:

$$\frac{\begin{array}{c}\neg maintained(\langle n, \Theta\rangle) \qquad NR = \bigcup_{n \rightsquigarrow n'} \langle n', \Theta\rangle \\ n \in N \qquad \langle n, \Theta\rangle \in is \qquad \langle n, \Theta\rangle \notin vs\end{array}}{\langle is, vs\rangle \overset{niv}{\rhd} \langle (is - \{\langle n, \Theta\rangle\}) \cup NR, vs \cup \{\langle n, \Theta\rangle\}\rangle}$$

Norm instance fulfilled:

$$\frac{\begin{array}{c}deactivated(n, \Theta') \\ n \in N \qquad \langle n, \Theta\rangle \in is \qquad \Theta' \subseteq \Theta\end{array}}{\langle is, fs\rangle \overset{nif}{\rhd} \langle is - \{\langle n, \Theta\rangle\}, fs \cup \langle n, \Theta\rangle\rangle}$$

Norm instance violation repaired:

$$\frac{\begin{array}{c}\langle n', \Theta\rangle \in fs \\ n, n' \in N \qquad n \rightsquigarrow n' \qquad \langle n, \Theta\rangle \in vs\end{array}}{\langle vs, rs\rangle \overset{nir}{\rhd} \langle vs - \{\langle n, \Theta\rangle\}, rs \cup \{\langle\langle n, \Theta\rangle, \langle n', \Theta\rangle\rangle\}\rangle}$$

Figure A.1: Inference rules for the transition relation $\rhd$

### A.2.2 Production System Semantics

Alvarez-Napagao *et al.* use a simplified version of the semantics for production systems introduced in [CKMM04].

Considering a set $\mathcal{P}$ of predicate symbols and an infinite set of variables $\mathcal{X}$:

- A fact is a grounded term $f \in \mathcal{T}(\mathcal{P})$.
- $\mathcal{WM}$ is the *working memory*, that is, a set of facts
- **if** $p, c$ **remove** $r$ **add** $a$ denotes a production rule. Formally: $p, c \Rightarrow r, a$.

A production rule consists of the following components:

- A set of positive and negative patterns $p = p^+ \cup p^-$. A pattern is a term $p_i \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. $\neg p_i$ denotes a negated pattern. $p^-$ denotes the set of all negate patterns and $p^+$ the rest of patterns.

- A proposition $c$ whose set of free variables is a subset of the pattern variables: $Var(c) \subseteq Var(p)$.
- A set $r$ of terms. Instances in the set will be removed from the working memory when the rule is fired, $r = \{r_i\}_{i \in I_r}$. $Var(r) \subseteq Var(p^+)$.
- A set $a$ of terms. Instances in the set will be added to the working memory when the rule is fired, $r = \{a_i\}_{i \in I_a}$. $Var(a) \subseteq Var(p)$.

The following definition introduces when a particular production rule matches to a set of facts, dismatches to a set of facts and is fireable with respect to a working memory. It also introduces how applying fireable rules on working memories lead to new working memories.

**Definition B 10** ***(Production rule semantics)***

*Given a set of positive patterns $p^+$, a set of facts $S$ and a substitution $\sigma$:*

*The set of positive patterns match the set of facts iff*

$$\forall p \in p^+ \; \exists t \in S : \sigma(p) = t.$$

*Given a set of negative patterns $p^-$, a set of facts $S$ and a substitution $\sigma$:*

*The set of negative patterns dismatches the set of facts iff*

$$\forall \neg p \in p^- \; \forall t \in S \; \forall \sigma : \sigma(p) \neq t.$$

*Given a production rule $p \Rightarrow r, a$ , a working memory $\mathcal{WM}$ and a substitution $\sigma$:*

*The production rule is $(\sigma, \mathcal{WM}')$-fireable iff the following conditions are met:*

*$p^+$ matches with $\mathcal{WM}'$*

*$p^-$ dismatches with $\mathcal{WM}$*

*$\mathcal{WM}'$ is a minimal subset of $\mathcal{WM}$*

$$\mathcal{T} \models \sigma(c)$$

*The application of a $(\sigma, \mathcal{WM}')$-fireable rule on a particular working memory $\mathcal{WM}$ leads to a new working memory $\mathcal{WM}''$ where:*

$$\mathcal{WM}'' = ((\mathcal{WM} - \sigma(r)) \cup \sigma(a))$$

Having introduced formally the concept of production rule, Alvarez-Napagao *et al.* introduce the concept of production system. Formally:

**Definition B 11** ***(Production System)***

*A general production system, denoted by $\mathcal{PS}$, is defined as:*

$$\mathcal{PS} = \langle \mathcal{P}, \mathcal{WM}_0, \mathcal{R} \rangle \text{ where } \mathcal{R} \text{ is the set of production rules over } (H) = \langle \mathcal{P}, X \rangle$$

They go on by introducing the formalisation of their normative monitor to a production system. In order to do so, they need to define several predicates to bind norms to their different conditions: *activation*, *mainteinance* and *deactivation*. They also need predicates to represent the normative state of the different norm instances: *violated*, *instantiated* and *fulfilled*. They also define a predicate to represent the arrival of new events: *event*, and a predicate to represent the fact a particular norm instance is a violation handling norm instance of a violated instance: *repair*. For handling the different DNF clauses, Alvarez-Napagao *et al.* define two additional predicates: *holds* and *has_clause*. Using these predicates, they are able to start instantiating a general production system in a production system for their Normative Monitor. They start by defining the set of predicates. Formally:

**Definition B 12** ***(Predicates of the Instantiated Production System)***
*Given an Institution $I$ of the form $I = \langle N, R, P, C, O \rangle$ the set of predicates in the instantiated production system is:*

$$(P)_i = O \cup \{activated, maintained, deactivated, violated, \\ instantiated, fulfilled, event, repair, holds, has\_clause, countsas\}$$

The initial working memory of the Instantiated Production System, denoted as $\mathcal{WM}_0$ should include the institutional specification. This can be achieved by including the formulas in the counts-as rules set, and the norms, in order to represent the possible instantiations of the predicate *holds* by using the predicate *has_clause*. To achieve this, Alvarez-Napagao *et al.* define bindings between the norms and the formulae available on the working memory. Formally:

**Definition B 13** ***(Norm-formula binding)***
*Given a norm $n$ of the form $n = \langle f_A, f_M, f_\delta, f_D, f_w, w \rangle$ its bindings are defined as:*

$$\mathcal{WM}_n := \{activation(n, f_A), mainteinance(n, f_M), deactivation(n, f_D)\}$$

*As it is assumed formulas in the set $wff(\mathcal{L}_O)$ are in DNF form, formulae can be interpreted as a set of conjunctive clauses, of which, only one of them holding true is necessary for the formula to hold true as well:*

$$r^h := has\_clause(f, f') \wedge holds(f', \Theta) \Rightarrow \emptyset, \{holds(f, \Theta)\}$$

By including the set of repair norms, via the *repair* predicate, and the counts-as definitions, via the *countsas* predicate, the initial working memory of the instantiated production system can be defined. Formally:

**Definition B 14** ***(Initial Working Memory of the Instantiated Production System)***
*Given an Institution $I$ of the form $I = \langle N, R, P, C, O \rangle$ the initial working memory of the instantiated production system for the Institution, denoted as $\mathcal{WM}_I$, is:*

$$\mathcal{WM}_I \bigcup_{n \rightsquigarrow n'}^{n \in N} repair(n, n') \cup$$

$$\bigcup_{n=\langle f_A,f_M,f_\delta,f_D,f_w,w\rangle\in N}(\mathcal{WM}_n\cup\mathcal{WM}_{f_A}\cup\mathcal{WM}_{f_M}\cup\mathcal{WM}_{f_D})\cup$$

$$\bigcup_{c=\langle\gamma_1,\gamma_2,s\rangle\in C}(\{countsas(\gamma_1,\gamma_2,s)\}\cup\mathcal{WM}_{\gamma_1}\cup\mathcal{WM}_s)$$

They define the rule for the detection of a holding formula as $r_f^{hc} = \lceil f\rceil \Rightarrow \emptyset, \{holds(f,\sigma)\}$ where $\lceil f\rceil$ denotes the propositional content of a formula $f \in wff((L)_O)$. Please, notice the formula only uses predicates from the Ontology $O$ and the logical connectives $\neg, \wedge$ and $\sigma$ as the substitution set of the activation rule. From this concept, Alvarez-Napagao *et al.* proceed by defining the set of rules for detection of holding formulae. Formally:

**Definition B 15** ***(Set of rules for detection of holding formulae for an institution)***
*Given an Institution I of the form $I = \langle N,R,P,C,O\rangle$ the set of rules for detection of holding formulae in the institution, denoted by $RI^{hc}$, is:*

$$RI^{hc} := \bigcup_{n=\langle f_A,f_M,f_\delta,f_D,f_w,w\rangle\in N}(\bigcup_{f\in\{f_A,f_M,fD\}} r_f^{hc})\cup\bigcup_{c=\langle\gamma_1,\gamma_2,s\rangle\in C}(\bigcup_{c\in\gamma_1} r_f^{hc})$$

Via the use of predicate *holds* Alvarez-Napagao *et al.* are able to provide a translation for the inference rules in *Figure* A.1 to production rules. Formally:

**Definition B 16** ***(Production rules for the inference rules)***
*Event processing:*

$$r^e = event(\alpha,p) \Rightarrow \emptyset, \{\lceil p\rceil\}$$

*Counts-as rule activation:*

$$r^{ca} = countas(\gamma_1,\gamma_2,c)\wedge holds(\gamma_1,\Theta)\wedge holds(\gamma_2,\Theta)\wedge holds(c,\Theta') \Rightarrow \emptyset, \{\Theta(\lceil\gamma_2\rceil)\}$$

*Counts-as rule deactivation:*

$$r^{cd} = countas(\gamma_1,\gamma_2,c)\wedge holds(\gamma_1,\Theta)\wedge holds(\gamma_2,\Theta)\wedge \neg holds(c,\Theta') \Rightarrow \{\Theta(\lceil\gamma_2\rceil)\}, \emptyset$$

*Norm Instantiation:*

$$r^{ni} = activation(n,f)\wedge holds(f,\Theta)\wedge\neg instantiated(n,\Theta)\wedge\neg repair(n',n) \Rightarrow \emptyset, \{instantiated(n,\Theta)\}$$

*Norm instance violation:*

$$r^{nv} = instantiated(n,\Theta)\wedge mainteinance(n,f)\wedge\neg holds(f,\Theta')\wedge repair(n,n'), \forall\Theta' : \Theta'\subseteq \Theta \Rightarrow \{instantiated(n,\Theta)\}, \{violated(n,\Theta), instantiated(n',\Theta)\}$$

*Norm instance fulfillment:*

$$r^{nf} = deactivation(n,f)\wedge instantiated(n,\Theta)\wedge holds(f,\Theta'), \forall\Theta' : \Theta'\subseteq\Theta \Rightarrow \{instantiated(n,\Theta)\}, \{fulfilled(n,\Theta)\}$$

*Norm instance violation repaired:*

$$r^{nr} = violated(n,\Theta)\wedge repair(n,n')\wedge fulfilled(n',\Theta') \Rightarrow \{violated(n,\Theta)\}, \emptyset$$

By using the above definitions, the set of rules for an Institution can be defined. Formally:

**Definition B 17** ***(Set of rules for an Institution)***
*Given an Institution $I$ of the form $I = \langle N, R, P, C, O\rangle$ the set of rules for the Institution $I$, denoted by $\mathcal{R}_I$ is:*

$$\mathcal{R}_I = R_I^{hc} \cup \{r^h, r^e, r^{ca}, r^{cd}, r^{ni}, r^{nv}, r^{nf}, r^{nr}\}$$

And finally, Alvarez-Napagao *et al.* define the production system for an institution. Formally:

**Definition B 18** ***(Production System for an Institution)***
*Given an Institution $I$ of the form $I = \langle N, R, P, C, O\rangle$ the production system for the Institution $I$, denoted by $\mathcal{PS}_I$, is:*

$$\mathcal{PS}_I = \langle \mathcal{P}_I, \mathcal{WM}_I, \mathcal{R}_I\rangle$$

### A.2.3 The implementation

Alvarez-Napagao *et al.* implement a prototype of their norm reasoner as a *DROOLS* program. *DROOLS* is an open source rule engine for declarative reasoning integrated in the Object-Oriented programming language Java. The rule engine implemented in *DROOLS* is based on the forward chaining inference RETE algorithm [For82]. Facts can be represented by adding them to the knowledge base as objects of the class *Predicate*. Predicates are imported dynamically from the standard description logic *OWL-DL* ontologies via the OWL2Java tool as subclasses of a particular *Predicate* class.

*DROOLS* programs can be initialised with a rule definition file. Both the working memory and the rule base can be modified at run-time by the Java process that is running the rule engine. They take advantage of this by keeping a fixed base, file with rules independent from the institution, and a parser for feeding the definitions that are dependant from the institution. The institutional definition Alvarez-Napagao *et al.* use is based on an extension of the XML language presented in [GR10]. The initial working memory can be automatically generated by inserting facts into the *DROOLS* knowledge base corresponding to the initial working memory of the institution as defined in *Definition*14.

The prototype implemented by Alvarez-Napagao *et al.* is available at `http://sf.net/projects/ict-alive` under *GPL* license.

# Bibliography


[AAB+03] T. Alsinet, C. Ansótegui, R. Béjar, C. Fernández, and F. Manyà, *Automated monitoring of medical protocols: a secure and distributed architecture*, Artificial Intelligence in Medicine **27** (2003), no. 3, 367–392.

[AAF14] I. Ayala, M. Amor, and L. Fuentes, *A model driven engineering process of platform neutral agents for ambient intelligence devices*, Autonomous agents and multi-agent systems **28** (2014), no. 2, 214–255.

[AAKS98] A. Alva-Argáez, A. Kokossis, and R. Smith, *Wastewater minimisation of industrial systems using an integrated approach*, Computers & Chemical Engineering **22** (1998), S741–S744.

[AÁNDVS10] H. Aldewereld, S. Álvarez-Napagao, F. Dignum, and J. Vázquez-Salceda, *Making norms concrete*, Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 807–814.

[ABA+98] T. Alsinet, R. Béjar, C. Ansótegui, C. Fernandez, and F. Manya, *Jafdis, a java framework for dialogical institution specification (extended version)*, Tech. report, Technical Report DIEI-98-RT-2, Universitat de Lleida, 1998. http://fermat. eup. udl. es/˜ cesar/recerca/jafdis_ev. ps. gz, 1998.

[ABGM01] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher, *Representation results for defeasible logic*, ACM Trans. Comput. Log. **2** (2001), no. 2, 255–287.

[AC10] R. Annichiarico and U. Cortés, *To Share or Not to Share SHARE-it Lessons Learnt*, eHealth, 2010, pp. 295–302.

[ACRV13] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, *A survey on ambient intelligence in healthcare*, Proceedings of the IEEE **101** (2013), no. 12, 2470–2494.

[AD11] H. Aldewereld and V. Dignum, *OperettA: Organization-oriented development environment*, Languages, Methodologies, and Development Tools for Multi-Agent Systems, Springer, 2011, pp. 1–18.

[AF09] M. Amor and L. Fuentes, *Malaca: A component and aspect-oriented agent architecture*, Information and Software Technology **51** (2009), no. 6, 1052–1065.

[AGHL09] G. Aucher, D. Grossi, A. Herzig, and E. Lorini, *Dynamic context logic*, Logic, Rationality, and Interaction, Springer, 2009, pp. 15–26.

[Ald07] H. Aldewereld, *Autonomy vs. conformity : an institutional perspective on norms and protocols*, Utrecht University, 2007.

[AN04] J. C. Augusto and C. D. Nugent, *The Use of Temporal Reasoning and Management of Complex Events in Smart Homes*, ECAI, 2004, pp. 778–782.

[ÁNAVSD10] S. Álvarez-Napagao, H. Aldewereld, J. Vázquez-Salceda, and F. Dignum, *Normative monitoring: Semantics and implementation*, COIN@AAMAS&MALLOW, 2010, pp. 321–336.

[ÁNCPVS09] S. Álvarez-Napagao, O. Cliffe, J. A. Padget, and J. Vázquez-Salceda, *Norms, Organisations and Semantic Web Services: The ALIVE approach*, Workshop on Coordination, Organization, Institutions and Norms at MALLOW'09 (2009), 1–2.

[APV+10] H. Aldewereld, J. Padget, W. Vasconcelos, J. Vázquez-Salceda, P. Sergeant, and A. Staikopoulos, *Adaptable, organization-aware, service-oriented computing*, Intelligent Systems **25** (2010), no. 4, 80–84.

[Axe86] R. Axelrod, *An evolutionary approach to norms*, American political science review **80** (1986), no. 04, 1095–1111.

[Bar99] M. Barbuceanu, *Role of obligations in multiagent coordination*, Applied Artificial Intelligence **13** (1999), no. 1-2, 11–38.

[BBB+05] L. Benedetti, F. Blumensaat, G. Bönisch, G. Dirckx, N. Jardin, P. Krebs, and P. Vanrolleghem, *Systems analysis of urban wastewater systems- two systematic approaches to analyse a complex system*, Water science and technology **52** (2005), no. 12, 171–180.

[BFPLC10] J. Bajo, J. A. Fraile, B. Pérez-Lancho, and J. M. Corchado, *The THOMAS architecture in home care scenarios: A case study*, Expert Systems with Applications **37** (2010), no. 5, 3986–3999.

[BLPN99] P. Baptiste, C. Le Pape, and W. Nuijten, *Satisfiability tests and time-bound adjustmentsfor cumulative scheduling problems*, Annals of Operations research **92** (1999), 305–333.

[BMJ09] R. Boulet, P. Mazzega, and B. Jouve, *Environmental, social and normative networks in the maelia platform*, Proc. ICAIL Worshop Legal & Decision Support Systems (2009), 83–93.

[BPG+04] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, *Tropos: An agent-oriented software development methodology*, Autonomous Agents and Multi-Agent Systems **8** (2004), no. 3, 203–236.

[BPN+09] L. Benedetti, P. Prat, I. Nopens, M. Poch, C. Turon, B. De Baets, and J. Comas, *A new rule generation method to develop a dss for integrated management at river basin scale*, 10th IWA conference on Instrumentation Control and Automation, 2009.

[BPR99] F. Bellifemine, A. Poggi, and G. Rimassa, *JADE–a FIPA-compliant agent framework*, Proceedings of PAAM, vol. 99, London, 1999, p. 33.

[BPR01] F. Bellifemine, A. Poggi, and G. Rimassa, *Jade: a fipa2000 compliant agent development environment*, Proceedings of the fifth international conference on Autonomous agents, ACM, 2001, pp. 216–217.

[BRT06] B. W. Brooks, T. M. Riley, and R. D. Taylor, *Water quality of effluent-dominated ecosystems: ecotoxicological, hydrological, and management considerations*, Hydrobiologia **556** (2006), no. 1, 365–379.

[BS05] D. Butler and M. Schutze, *Integrating simulation models with a view to optimal control of urban wastewater systems*, Environmental Modelling & Software **20** (2005), 415–426.

[BS08] S. Bromuri and K. Stathis, *Situating cognitive agents in golem*, Engineering environment-mediated multi-agent systems, Springer, 2008, pp. 115–134.

[BvdTV08] G. Boella, L. van der Torre, and H. Verhagen, *Introduction to the special issue on normative multiagent systems*, Autonomous Agents and Multi-Agent Systems **17** (2008), no. 1, 1–10.

[CAGP13] L. Corominas, V. Acuña, A. Ginebreda, and M. Poch, *Integration of freshwater environmental policies and wastewater treatment plant management*, Science of the Total Environment **445** (2013), 185–191.

[Cas98] C. Castelfranchi, *Modelling social action for AI agents*, Artificial Intelligence **103** (1998), no. 1, 157–182.

[CBA08] J. M. Corchado, J. Bajo, and A. Abraham, *GerAmi: Improving healthcare delivery in geriatric residences*, Intelligent Systems, IEEE **23** (2008), no. 2, 19–25.

[CDVP07a] O. Cliffe, M. De Vos, and J. Padget, *Answer set programming for representing and reasoning about virtual institutions*, Computational Logic in Multi-Agent Systems, Springer, 2007, pp. 60–79.

[CDVP07b] O. Cliffe, M. De Vos, and J. Padget, *Specifying and reasoning about multiple institutions*, Coordination, Organizations, Institutions, and Norms in Agent Systems II, Springer, 2007, pp. 67–85.

[Che93] A. M. K. Cheng, *Parallel execution of real-time rule-based systems*, IPPS, 1993, pp. 779–786.

[CKMM04] H. Cirstea, C. Kirchner, M. Moossen, and P.-E. Moreau, *Production systems and rete algorithm formalisation*, Tech. report, inria, 2004.

[CLSRAE09] J. Campos, M. López-Sánchez, J. A. Rodríguez-Aguilar, and M. Esteva, *Formalising situatedness and adaptation in electronic institutions*, Coordination, Organizations, Institutions and Norms in Agent Systems IV, Lecture Notes in Computer Science, Springer Berlin / Heidelberg **5428** (2009), 126–139.

[CM08] M. Charfeddine and B. Montreuil, *Toward a conceptual agent-based framework for modelling and simulation of distributed healthcare delivery systems*, CIRRELT, 2008.

[CMA01] L. M. Camarinha-Matos and H. Afsarmanesh, *Virtual enterprise modeling and support infrastructures: applying multi-agent system approaches*, Multi-agent systems and applications, Springer, 2001, pp. 335–364.

[CO09] H. L. Cardoso and E. Oliveira, *A context-based institutional normative environment*, Coordination, Organizations, Institutions and Norms in Agent Systems IV, Springer, 2009, pp. 140–155.

[COML12] M. Croitoru, N. Oren, S. Miles, and M. Luck, *Graphical norms via conceptual graphs*, Knowledge-Based Systems **29** (2012), 31–43.

[COS02] A. Cesta, A. Oddi, and S. F. Smith, *A Constraint-Based Method for Project Scheduling with Time Windows*, Journal of Heuristics **8** (2002), 109–136.

[Cou91] Council Of European Communities, *European communities. directive concerning urban waste water treatment (91/271/eec)*, vol. 30, Officia/Jour˜ a/L135/40, 1991.

[CP09] U. Cortés and M. Poch, *Advanced agent-based environmental management systems*, Springer, 2009.

[CPRRC02] J. Comas, M. Poch, I. Rodríguez-Roda, and U. Cortés, *Eleven years of experience in designing and building real environmental decision support systems: what have we learnt?*, Universitat de Girona, 2002.

[CSMC+00] U. Cortés, M. Sànchez-Marrè, L. Ceccaroni, I. R-Roda, and M. Poch, *Artificial intelligence and environmental decision support systems*, Applied Intelligence **13** (2000), no. 1, 77–91.

[Dav00] P. Davidsson, *Emergent societies of information agents*, In Fourth International Workshop on Cooperative Information Agents (CIA'2000, Springer Verlag, 2000, pp. 143–153.

[DBDM04] F. Dignum, J. Broersen, V. Dignum, and J.-J. C. Meyer, *Meeting the deadline: Why, when and how*, FAABS, 2004, pp. 30–40.

[DCT+09] F. Devesa, J. Comas, C. Turon, A. Freixó, F. Carrasco, and M. Poch, *Scenario analysis for the role of sanitation infrastructures in integrated urban wastewater management*, Environmental modelling & software **24** (2009), no. 3, 371–380.

[DDPVS09] F. Dignum, V. Dignum, J. Padget, and J. Vázquez-Salceda, *Organizing web services to develop dynamic, flexible, distributed systems*, Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, ACM, 2009, pp. 225–234.

[Dig99] F. Dignum, *Autonomous agents with norms*, Artif. Intell. Law **7** (1999), no. 1, 69–79.

[Dig04] V. Dignum, *A model for organizational interaction: Based on agents, founded in logic*, SIKS Dissertation Series 2204-1 (U. University, ed.), Utrecht University, 2004.

[DPS03] J. Delgado, J. M. Pujol, and R. Sangüesa, *Emergence of coordination in scale-free networks*, Web Intelligence and Agent Systems **1** (2003), no. 2, 131–138.

[DRSG+14] M. Di Rocco, S. Sathyakeerthy, J. Grosinger, F. Pecora, A. Saffiotti, M. Bonaccorsi, F. Cavallo, R. Limosani, A. Manzi, G. Teti, et al., *A planner for ambient assisted living: From high-level reasoning to low-level robot execution and back*, 2014 AAAI Spring Symposium Series, 2014.

[Dur04] A. Duran, *Dinámica de los usos y la gestión del agua en tiquipaya y colcapirhua*, Proyecto NEGOWAT (2004), 1–14.

[DVSD04] V. Dignum, J. Vázquez-Salceda, and F. Dignum, *A Model of Almost Everything: Norms, Structure and Ontologies in Agent Organizations*, AAMAS, 2004, pp. 1498–1499.

[DWS01] S. A. DeLoach, M. F. Wood, and C. H. Sparkman, *Multiagent systems engineering*, International Journal of Software Engineering and Knowledge Engineering **11** (2001), no. 03, 231–258.

[DWX02] V. Dignum, H. Weigand, and L. Xu, *Agent societies: towards frameworks-based design*, Agent-Oriented Software Engineering II, Springer, 2002, pp. 33–49.

[EDLCS02] M. Esteva, D. De La Cruz, and C. Sierra, *ISLANDER: an electronic institutions editor*, Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3, ACM, 2002, pp. 1045–1052.

[EFG+02] V. Erbe, T. Frehmann, W. Geiger, P. Krebs, J. Londong, K. Rosenwinkel, and K. Seggelke, *Integrated modelling as an analysing and optimisation tool for urban watershed management*, Water science and technology **46** (2002), no. 6-7, 141–150.

[EHRLR80] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and R. Reddy, *The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty*, ACM Comput. Surv. **12** (1980), no. 2, 213–253.

[ERSL02] V. Erbe, L. Risholt, W. Schilling, and J. Londong, *Integrated modelling for analysis and optimisation of wastewater systems–the odenthal case*, Urban Water **4** (2002), no. 1, 63–71.

[ES05] V. Erbe and M. Schtze, *An integrated modelling concept for immission-based management of sewer system, wastewater treatment plant and river*, Water Science & Technology **52** (2005), no. 5, 95–103.

[FD00] J. Fox and S. Das, *Safe and sound*, Artificial Intelligence in Hazardous Applications (2000), 30–7.

[FGM04] J. Ferber, O. Gutknecht, and F. Michel, *From agents to organizations: An organizational view of multi-agent systems*, Agent-Oriented Software Engineering IV (P. Giorgini, J. Müller, and J. Odell, eds.), Lecture Notes in Computer Science, vol. 2935, Springer Berlin / Heidelberg, 2004, 10.1007/978-3-540-24620-6-15, pp. 443–459.

[FIP00] FIPA: Foundation for Intelligent Physical Agents, *Iterated contract net interaction protocol specification http://www. fipa. org/specs/fipa00030*, vol. 90, PC00030D. pdf, 2000.

[For82] C. Forgy, *Rete: A fast algorithm for the many patterns/many objects match problem*, Artif. Intell. **19** (1982), no. 1, 17–37.

[Fri77] L. M. Friedman, *Law and society: An introduction*, Prentice Hall, 1977.

[FT98] D. Fitoussi and M. Tennenholtz, *Minimal social laws*, AAAI/IAAI, 1998, pp. 26–31.

[FWJ04] S. S. Fatima, M. Wooldridge, and N. R. Jennings, *An agenda-based framework for multi-issue negotiation*, Artificial Intelligence **152** (2004), no. 1, 1–45.

[Gäc07] S. Gächter, *Conditional cooperation: Behavioral regularities from the lab and the field and their policy implications*, na, 2007.

[GAD07] D. Grossi, H. Aldewereld, and F. Dignum, *Ubi lex, ibi poena : Designing norm enforcement in e-institutions*, Coordination, Organizations, Institutions, and Norms in Agent Systems II (P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, eds.), Lecture Notes in Computer Science, vol. 4386, Springer Berlin / Heidelberg, 2007, pp. 101–114.

[GAVSD06] D. Grossi, H. Aldewereld, J. Vázquez-Salceda, and F. Dignum, *Ontological aspects of the implementation of norms in agent-based electronic institutions*, Computational & Mathematical Organization Theory **12** (2006), no. 2-3, 251–275.

[GBRHP12] M. Garrido-Baserba, R. Reif, F. Hernández, and M. Poch, *Implementation of a knowledge-based methodology in a decision support system for the design of suitable wastewater treatment process flow diagrams*, Journal of environmental management **112** (2012), 384–391.

[GCNRA05] A. García-Camino, P. Noriega, and J. A. Rodríguez-Aguilar, *Implementing norms in electronic institutions*, AAMAS (F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, eds.), ACM, 2005, pp. 667–673.

[Gen] Generalitat de Catalunya, *Psaru 2005*, `http://aca-web.gencat.cat/aca/appmanager/aca/aca?_nfpb=true&_pageLabel=P1204654461208200526170&profileLocale=es`, Accessed: 2015-03-30.

[GFK+88] A. Gupta, C. Forgy, D. Kalp, A. Newell, and M. Tambe, *Parallel ops5 on the encore multimax*, ICPP (1), 1988, pp. 71–280.

[GGBN13] A. Garrido, A. Giret, V. Botti, and P. Noriega, *mwater, a case study for modeling virtual markets*, Agreement Technologies, Springer, 2013, pp. 565–582.

[Gov] S. Government, *Real decreto-ley 11/1995*, `http://www.boe.es/boe/dias/1995/12/30/pdfs/A37517-37519.pdf`, Accessed: 2015-03-30.

[Gov05] G. Governatori, *Representing business contracts in ruleml*, International Journal of Cooperative Information Systems **14** (2005), no. 02n03, 181–216.

[GPGS+03] L. Godo, J. Puyol-Gruart, J. Sabater, V. Torra, P. Barrufet, and X. Fabregas, *A multi-agent system approach for monitoring the prescription of restricted use antibiotics*, Artificial Intelligence in Medicine **27** (2003), no. 3, 259–282.

[GR08a] G. Governatori and A. Rotolo, *Changing legal systems: Abrogation and annulment part i: Revision of defeasible theories*, DEON, 2008, pp. 3–18.

[GR08b] G. Governatori and A. Rotolo, *Changing legal systems: Abrogation and annulment. part ii: Temporalised defeasible logic*, NORMAS, 2008, pp. 112–127.

[GR10] G. Governatori and A. Rotolo, *Norm compliance in business process modeling*, RuleML, 2010, pp. 194–209.

[Gro07] D. Grossi, *Designing invisible handcuffs: Formal investigations in institutions and organizations for multi-agent systems*, vol. 16, Utrech University, 2007.

[GRS05] G. Governatori, A. Rotolo, and G. Sartor, *Temporalised normative positions in defeasible logic*, ICAIL, 2005, pp. 25–34.

[GSANVS11] I. Gómez-Sebastià, S. Alvarez-Napagao, and J. Vázquez-Salceda, *A distributed norm compliance model.*, CCIA **232** (2011), 110–119.

[GSANVS13] I. Gómez-Sebastià, S. Alvarez-Napagao, and J. Vázquez-Salceda, *Towards heuristic based mobility policy optimisation.*, CCIA, 2013, pp. 297–300.

[GSÁNVSF12] I. Gómez-Sebastià, S. Álvarez-Napagao, J. Vázquez-Salceda, and L. O. Felipe, *Towards runtime support for norm change from a monitoring perspective.*, AT, 2012, pp. 71–85.

[GSGGÁN11] I. Gómez-Sebastià, D. Garcia-Gasulla, and S. Álvarez-Napagao, *Society of situated agents for adaptable eldercare*, ERCIM news (2011), no. 87, 23–24.

[GSGGAN+12] I. Gómez-Sebastià, D. Garcia-Gasulla, S. Alvarez-Napagao, J. Vázquez-Salceda, and U. Cortés, *Towards an implementation of a social electronic reminder for pills*, VII Workshop on Agents Applied in Health Care, A2HC 2012 (2012), 61–70.

[GSGGB+12] I. Gómez-Sebastia, D. Garcia-Gasulla, C. Barrue, J. Vázquez-Salceda, and U. Cortés, *Alive meets share-it: An agent-oriented solution to model organisational and normative requirements in assistive technologies*, Electronic Healthcare, Springer, 2012, pp. 319–326.

[H+69] H. R. Hamilton et al., *Systems simulation for regional analysis: an application to river-basin planning.*, Cambridge, Mass./London: Mass. Inst. Technol., 1969.

[HCDVP10] L. Hopton, O. Cliffe, M. De Vos, and J. Padget, *Instql: A query language for virtual institutions using answer set programming*, Computational Logic in Multi-Agent Systems, Springer, 2010, pp. 102–121.

[HKM+04] K. Z. Haigh, L. M. Kiff, J. Myers, V. Guralnik, C. W. Geib, J. Phelps, and T. Wagner, *The Independent LifeStyle Assistant (I.L.S.A.): AI Lessons Learned*, In The Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI-04, 2004, pp. 25–29.

[HND+07] J. Hallberg, C. D. Nugent, R. J. Davies, K. Synnes, M. P. Donnelly, D. Finlay, and M. Mulvenna, *HomeRuleML-a model for the exchange of decision support rules within smart environments*, Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on, IEEE, 2007, pp. 513–520.

[HO03] W. Health Organization", *Adherence to long-term therapies. Evidence for action*, Tech. report, World Health Organization, 2003.

[HPSB+04] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, et al., *Swrl: A semantic web rule language combining owl and ruleml*, W3C Member submission **21** (2004), 79.

[Hud93] R. E. Hudec, *Enforcing international trade law: The evolution of the modern gatt legal system*, vol. 7, Butterworth Legal Publishers Salem, NH, 1993.

[HZ00] G. Huang and Q. Zhao, *Multi-objective solutions for coordinating auctions of different commodities in power markets*, Power Engineering Society Summer Meeting, 2000. IEEE, vol. 4, IEEE, 2000, pp. 2132–2137.

[IHK07] T. Ito, H. Hattori, and M. Klein, *Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces.*, IJCAI, vol. 7, 2007, pp. 1347–1352.

[Inn12] Innovators In Health, *uBox, a Smart Pillbox for TB Treatment*, Tech. report, MIT, 2012.

[JA] J. Jiang and H. Aldewereld, *Modeling and detecting norm conflicts in regulated organizations*.

[JAD] JADE, *Java agent development environment*, http://sharon.cslet.it/project/jade.

[JADT14] J. Jiang, H. Aldewereld, V. Dignum, and Y.-H. Tan, *Compliance checking of organizational interactions*, ACM Transactions on Management Information Systems (TMIS) **5** (2014), no. 4, 23.

[Jap] *Japan may pick robots over immigrants*, `http://news.bbc.co.uk/2/hi/business/8687196.stm`, Accessed: 2015-01-07.

[JAP13] A. J. Jones, A. Artikis, and J. Pitt, *The design of intelligent socio-technical systems*, Artificial Intelligence Review **39** (2013), no. 1, 5–20.

[JDT12] J. Jiang, V. Dignum, and Y.-H. Tan, *An agent-based inter-organizational collaboration framework: Opera+*, Coordination, Organizations, Institutions, and Norms in Agent System VII, Springer, 2012, pp. 58–74.

[Jen01] N. R. Jennings, *An agent-based approach for building complex software systems*, Communications of the ACM **44** (2001), no. 4, 35–41.

[Jen13] K. Jensen, *Coloured petri nets: basic concepts, analysis methods and practical use*, vol. 1, Springer Science & Business Media, 2013.

[JS96] A. J. Jones and M. Sergot, *A formal characterisation of institutionalised power*, Logic Journal of IGPL **4** (1996), no. 3, 427–443.

[JSW98] N. R. Jennings, K. Sycara, and M. Wooldridge, *A roadmap of agent research and development*, Autonomous agents and multi-agent systems **1** (1998), no. 1, 7–38.

[KBLA+09] M. N. Kamel Boulos, R. C. Lou, A. Anastasiou, C. D. Nugent, J. Alexandersson, G. Zimmermann, U. Cortés, and R. Casas, *Connectivity for healthcare and well-being management: examples from six European projects*, International Journal of Environmental Research and Public Health **6** (2009), no. 7, 1947–1971.

[KBS+13] Ö. Kafalı, S. Bromuri, M. Sindlar, T. van der Weide, E. Aguilar Pelaez, U. Schaechtle, B. Alves, D. Zufferey, E. Rodriguez-Villegas, M. I. Schumacher, et al., *Commodity 12: A smart e-health environment for diabetes management*, Journal of Ambient Intelligence and Smart Environments **5** (2013), no. 5, 479–502.

[Kop11] H. Kopetz, *Internet of things*, Real-time systems, Springer, 2011, pp. 307–323.

[KPJ+08] J. A. Kientz, S. N. Patel, B. Jones, E. Price, E. D. Mynatt, and G. D. Abowd, *The georgia tech aware home*, CHI'08 extended abstracts on Human factors in computing systems, ACM, 2008, pp. 3675–3680.

[LBDV+13] T. Li, T. Balke, M. De Vos, K. Satoh, and J. Padget, *Detecting conflicts in legal systems*, New Frontiers in Artificial Intelligence, Springer, 2013, pp. 174–189.

[LE80] V. Lesser and L. D. Erman, *Distributed interpretation: A model and experiment*, IEEE Trans. Computers **29** (1980), no. 12, 1144–1163.

[Les90] V. Lesser, *An overview of dai: Viewing distributed ai as distributed search*, Journal of Japanese Society for Artificial Intelligence **5** (1990), 392–400.

[LGC06] E. López-Gunn and L. M. Cortina, *Is self-regulation a myth? case study on spanish groundwater user associations and the role of higher-level authorities*, Hydrogeology Journal **14** (2006), no. 3, 361–379.

[LL03] F. López-López, *Social power and norms: Impact on agent behavior*, Ph.D. thesis, PhD thesis, University of Southampton, 2003.

[LLLd07] F. López-López, M. Luck, and M. d'Inverno, *A normative framework for agent-based systems*, Normative Multi-agent Systems (Dagstuhl, Germany) (G. Boella, L. van der Torre, and H. Verhagen, eds.), Dagstuhl Seminar Proceedings, no. 07122, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

[Mat95] M. J. Matarić, *Issues and approaches in the design of collective autonomous agents*, Robotics and autonomous systems **16** (1995), no. 2, 321–331.

[MBB+11] E. Marengo, M. Baldoni, C. Baroglio, A. K. Chopra, V. Patti, and M. P. Singh, *Commitments with regulations: reasoning about safety and control in regula*, AAMAS (L. Sonenberg, P. Stone, K. Tumer, and P. Yolum, eds.), IFAAMAS, 2011, pp. 467–474.

[MBD+11] J. Murillo, D. Busquets, J. Dalmau, B. López, V. Muñoz, and I. Rodríguez-Roda, *Improving urban wastewater management through an auction-based management of discharges*, Environmental Modelling & Software **26** (2011), no. 6, 689–696.

[MBH+04] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, et al., *Owl-s: Semantic markup for web services*, W3C member submission **22** (2004), 2007–04.

[MC94] T. W. Malone and K. Crowston, *The interdisciplinary study of coordination*, ACM Computing Surveys (CSUR) **26** (1994), no. 1, 87–119.

[McC12] S. McCallum, *Gamification and serious games for personalized health*, Studies in health technology and informatics **177** (2012), 85–96.

[MHS+01] J. Meirlaen, B. Huyghebaert, F. Sforzi, L. Benedetti, and P. Vanrolleghem, *Fast, simultaneous simulation of the integrated urban wastewater system using mechanistic surrogate models*, Water science and technology **43** (2001), no. 7, 301–310.

[MLSRA+13] J. Morales, M. López-Sanchez, J. A. Rodriguez-Aguilar, M. Wooldridge, and W. Vasconcelos, *Automated synthesis of normative systems*, Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 483–490.

[MLSRA+14] J. Morales, M. López-Sanchez, J. A. Rodriguez-Aguilar, M. Wooldridge, and W. Vasconcelos, *Minimality and simplicity in the on-line automated synthesis of normative systems*, Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 109–116.

[MLSRA+15] J. Morales, M. López-Sánchez, J. A. Rodriguez-Aguilar, M. Wooldridge, and W. Vasconcelos, *Synthesising liberal normative systems*, Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 433–441.

[MMB+07] V. Muñoz, J. Murillo, D. Busquets, B. López, R. K. Dash, N. R. Jennings, D. C. Parkes, S. Singh, M. Tiwara, A. Larsen, et al., *Improving water quality by coordinating industries schedules and treatment plants*, AAMAS workshop on Coordinating Agents Plans ans Schedules (CAPS), 2007, pp. 1–8.

[MNL+08] C. Makropoulos, K. Natsis, S. Liu, K. Mittas, and D. Butler, *Decision support for sustainable option selection in integrated urban water management*, Environmental Modelling & Software **23** (2008), no. 12, 1448–1460.

[Mor13] J. Moreno, *SPiDer: Smart Pill Dispenser*, Master's thesis, Universitat Politècnica de Catalunya, March 2013.

[MSA08] P. Mukherjee, S. Sen, and S. Airiau, *Norm emergence under constrained interactions in diverse societies*, Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2, International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 779–786.

[MT95] Y. Moses and M. Tennenholtz, *Artificial social systems*, Computers and AI **14** (1995), 533–562.

[MVAV02] J. Meirlaen, J. Van Assel, and P. Vanrolleghem, *Real time control of the integrated urban wastewater system using simultaneously simulating surrogate models*, Water science and technology **45** (2002), no. 3, 109–116.

[Nat07] National Council on Patient Information and Education, *Enhancing prescription medicine adherence. a national action plan*, 2007.

[NFKG07] M. Niemelä, R. G. Fuentetaja, E. Kaasinen, and J. L. Gallardo, *Supporting Independent Living of the Elderly with Mobile-Centric Ambient Intelligence: User Evaluation of Three Scenarios*, AmI, 2007, pp. 91–107.

[NM03] J. Nealon and A. Moreno, *Agent-based applications in Health Care*, Applications of software agent technology in the health care domain, Springer, 2003, pp. 3–18.

[Nor90] D. C. North, *Institutions, Institutional Change and Economic Performance (Political Economy of Institutions and Decisions)*, Cambridge University Press, October 1990.

[NSS94] E. Nuutila and E. Soisalon-Soininen, *On finding the strongly connected components in a directed graph*, Information Processing Letters **49** (1994), no. 1, 9–14.

[OPVS+09] N. Oren, S. Panagiotidi, J. Vázquez-Salceda, S. Modgil, M. Luck, and S. Miles, *Towards a formalisation of electronic contracting environments*, Coordination, Organizations, Institutions and Norms in Agent Systems IV (J. Hübner, E. Matson, O. Boissier, and V. Dignum, eds.), Lecture Notes in Computer Science, vol. 5428, Springer Berlin / Heidelberg, 2009, 10.1007/978-3-642-00443-8-11, pp. 156–171.

[Ost90] E. Ostrom, *Governing the commons: The evolution of institutions for collective action*, Cambridge university press, 1990.

[PANVS14] S. Panagiotidi, S. Alvarez-Napagao, and J. Vázquez-Salceda, *Towards the norm-aware agent: bridging the gap between deontic specifications and practical mechanisms for norm monitoring and norm-aware planning*, Coordination, Organizations, Institutions, and Norms in Agent Systems IX, Springer, 2014, pp. 346–363.

[PBM14] J. Pitt, D. Busquets, and S. Macbeth, *Distributive justice for self-organised common-pool resource management*, ACM Transactions on Autonomous and Adaptive Systems (TAAS) **9** (2014), no. 3, 14.

[PCJ04] S. Paurobally, J. Cunningham, and N. Jennings, *Verifying the contract net protocol: a case study in interaction protocol and agent communication semantics*, Proceedings of 2nd International Workshop on Logic and Communication in Multi-Agent Systems, 2004, pp. 98–117.

[PCRR+04] M. Poch, J. Comas, I. Rodríguez-Roda, M. Sànchez-Marrè, and U. Cortés, *Designing and building real environmental decision support systems*, Environmental Modelling & Software **19** (2004), no. 9, 857–873.

[PN88] W. D. Potter and D. Nute, *d-kdl: An eds environment incorporating defeasible reasoning*, DS-3, 1988, pp. 529–538.

[Pop12] Population Division UN Department of Economic Social Affairs, *Population ageing and development: Ten years after Madrid*, Tech. Report 2012/4, United Nations, December 2012.

[PS12] J. Pitt and J. Schaumeier, *Provision and appropriation of common-pool resources without full disclosure*, PRIMA 2012: Principles and Practice of Multi-Agent Systems, Springer, 2012, pp. 199–213.

[PW03] L. Padgham and M. Winikoff, *Prometheus: A methodology for developing intelligent agents*, Agent-Oriented Software Engineering III, Springer, 2003, pp. 174–185.

[RAK+98] W. Rauch, H. Aalderink, P. Krebs, W. Schilling, and P. Vanrolleghem, *Requirements for integrated wastewater models-driven by receiving water objectives*, Water Science & Technology **38** (1998), no. 11, 97–104.

[Res66] N. Rescher, *Distributive justice: A constructive critique of the utilitarian theory of distribution*, Irvington Publishers, 1966.

[RSSMAC06] T. Rendón-Sallard, M. Sànchez-Marrè, M. Aulinas, and J. Comas, *Designing a Multi-Agent system to simulate scenarios for decision-making in river basin systems*, Proceedings of the 2006 Conference on Artificial Intelligence Research and Development, IOS Press, 2006, pp. 291–298.

[SA07] S. Sen and S. Airiau, *Emergence of norms through social learning.*, IJCAI, vol. 1507, 2007, p. 1512.

[Sad11] F. Sadri, *Ambient intelligence: A survey*, ACM Computing Surveys (CSUR) **43** (2011), no. 4, 36.

[SBG+08] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, J. Rashid, B. Seo, and Y.-J. Cho, *The PEIS-ecology project: vision and results*, Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, IEEE, 2008, pp. 2329–2335.

[SBLB01] T. Schultz, A. Bc, D. Langer, and V. Bonnet, *Online dispute resolution: The state of the art and the issues*, Negotiation Journal **11** (2001), no. 2, 117–121.

[SCC+03] M. Schütze, A. Campisano, H. Colas, P. Vanrolleghem, and W. Schilling, *Real-time control of urban water systems*, Proceedings of the International Conference on Pumps, Electromechanical Devices and Systems Applied to Urban Water Management, Citeseer, 2003, pp. 1–19.

[Sch08] G. Schäfer, *Europe in figures. Eurostat statistical yearbook*, 2008.

[SH06] T. Schmitt and W. Huber, *The scope of integrated modelling: system boundaries, sub-systems, scales and disciplines*, Water Science & Technology **54** (2006), no. 6-7, 405–413.

[SHWS09] A. Smajgl, S. Heckbert, J. Ward, and A. Straton, *Simulating impacts of water trading in an institutional perspective*, Environmental Modelling & Software **24** (2009), no. 2, 191–201.

[Sin96] M. P. Singh, *Commitments in the architecture of a limited, rational agent*, PRICAI Workshop on Intelligent Agent Systems (L. Cavedon, A. S. Rao, and W. Wobcke, eds.), Lecture Notes in Computer Science, vol. 1209, Springer, 1996, pp. 72–87.

[Smi80] R. G. Smith, *The contract net protocol: High-level communication and control in a distributed problem solver*, IEEE Transactions on computers (1980), no. 12, 1104–1113.

[Sou08] T. Sourdin, *Alternative dispute resolution*, Thomson Lawbook Company, 2008.

[SRVK05] K. Seggelke, K. Rosenwinkel, P. A. Vanrolleghem, and P. Krebs, *Integrated operation of sewer system and wwtp by simulation-based control of the wwtp inflow*, Water Science & Technology **52** (2005), no. 5, 195–203.

[SS04] A. Sturm and O. Shehory, *A comparative evaluation of agent-oriented methodologies*, Methodologies and software engineering for agent Systems, Springer, 2004, pp. 127–149.

[SS09] F. Sadri and K. Stathis, *Ambient Intelligence*, Encyclopedia of Artificial Intelligence, Information Science Reference-Imprint of: IGI Publishing, 2009, pp. 85–91.

[ST95] Y. Shoham and M. Tennenholtz, *On social laws for artificial agent societies: Off-line design*, Artificial Intelligence **73** (1995), 231–252.

[Syc98] K. P. Sycara, *Multiagent systems*, AI magazine **19** (1998), no. 2, 79.

[TCM+06] P. Tolchinsky, U. Cortés, S. Modgil, F. Caballero, and A. López-Navidad, *Increasing human-organ transplant availability: Argumentation-based agent deliberation*, Intelligent Systems, IEEE **21** (2006), no. 6, 30–37.

[Tho97] M. Thobanl, *Formal water markets: Why, when, and how to introduce tradable water rights*, The World Bank Research Observer **12** (1997), no. 2, 161–179.

[UBSA10] V. Urovi, S. Bromuri, K. Stathis, and A. Artikis, *Initial steps towards run-time support for norm-governed systems*, COIN@AAMAS&MALLOW, 2010, pp. 268–284.

[UM15] E. Ullmann-Margalit, *The emergence of norms*, OUP Oxford, 2015.

[Uni04] United Nations Department of Economic, *World population to 2030*, vol. 236, United Nations Publications, 2004.

[Van96] P. Vanrolleghem, *Evaluation of design and operation of the sewage transport and treatment system by an EQO/EQS based analysis of the receiving water immission characteristics*, Proceedings of the WEF Conference Urban Wet Weather Pollution., 1996, pp. 14.35–14.46.

[VCP12] M. Verdaguer, N. Clara, and M. Poch, *Ant colony optimization-based method for managing industrial influents in wastewater systems*, AIChE Journal **58** (2012), no. 10, 3070–3079.

[VF13] O. Vermesan and P. Friess, *Internet of Things: converging technologies for smart environments and integrated ecosystems*, River Publishers, 2013.

[Vig07] L. L. Viguier, *Co2 emissions: an allowance auction system should be adopted*, Le Temps, 2007.

[VS03] J. Vázquez-Salceda, *The role of norms and electronic institutions in multi-agent systems applied to complex domains. the harmonia framework: Thesis*, AI Commun. **16** (2003), no. 3, 209–212.

[VSCP+03] J. Vázquez-Salceda, U. Cortés, J. Padget, A. López-Navidad, and F. Caballero, *The organ allocation process: a natural extension of the cARREL Agent-mediated Electronic Institution*, AI Communications **16** (2003), no. 3, 153–165.

[VSMS11] D. Villatoro, J. Sabater-Mir, and S. Sen, *Social instruments for robust convention emergence*, IJCAI, vol. 11, 2011, pp. 420–425.

[VSVP+10] J. Vázquez-Salceda, W. W. Vasconcelos, J. Padget, F. Dignum, S. Clarke, and M. Palau Roig, *ALIVE: an agent-based framework for dynamic and robust service-oriented applications*, Proceedings of the $9^{th}$ International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 1637–1638.

[WD01] M. F. Wood and S. A. DeLoach, *An overview of the multiagent systems engineering methodology*, Agent-Oriented Software Engineering, Springer, 2001, pp. 207–221.

[Wei99a] G. Weiss (ed.), *Multiagent systems: A modern approach to distributed artificial intelligence*, MIT Press, 1999.

[Wei99b] G. Weiss, *Multiagent systems: a modern approach to distributed artificial intelligence*, MIT press, 1999.

[WJ95] M. Wooldridge and N. R. Jennings, *Intelligent agents: Theory and practice*, The Knowledge Engineering Review (1995), no. 2, 115–152.

[WJK00a] M. Wooldridge, N. R. Jennings, and D. Kinny, *The gaia methodology for agent-oriented analysis and design*, Autonomous Agents and Multi-Agent Systems **3** (2000), 285–312, 10.1023/A:1010071910869.

[WJK00b] M. Wooldridge, N. R. Jennings, and D. Kinny, *The gAIA methodology for agent-oriented analysis and design*, Autonomous Agents and Multi-Agent Systems **3** (2000), no. 3, 285–312.

[Woo97] M. Wooldridge, *Agent-based software engineering*, IEE Proceedings-software **144** (1997), no. 1, 26–37.

[ZJW03] F. Zambonelli, N. R. Jennings, and M. Wooldridge, *Developing multiagent systems: The Gaia methodology*, ACM Transactions on Software Engineering and Methodology (TOSEM) **12** (2003), no. 3, 317–370.

[ZO04] F. Zambonelli and A. Omicini, *Challenges and research directions in agent-oriented software engineering*, Autonomous Agents and Multi-Agent Systems **9** (2004), no. 3, 253–283.

# Publications

This annex presents the list of conference papers and scientific journal publications of the research period corresponding to the development of this thesis. The details of the papers as well as their digital versions are available on my personal web page[1]

JOURNAL PUBLICATIONS

[1] Gómez-Sebastià, I., Garcia Gasulla, D., and Alvarez Napagao, S. (2011). Society of situated agents for adaptable eldercare. "ERCIM news", 01 October 2011, num. 87, p. 23-24.

[2] Gómez-Sebastià, I., Moreno, J., Alvarez Napagao, S., Garcia Gasulla, D., Barrué, C., and Cortés U. Situated agents and humans in social interaction for elderly healthcare From Coaalas to AVICENA, Journal of Medical Systems Special Issue, Agent-Empowered HealthCare Systems, 2014. DOI: 10.1007/s10916-015-0371-7

CONFERENCE AND WORKSHOP PUBLICATIONS RELATED TO THE PHD THESIS

[3] Gómez-Sebastià, I., Palau, M., Nieves, J. C., Vázquez-Salceda, J. and Ceccaroni, L. (2009, January). Dynamic orchestration of distributed services on interactive community displays: The ALIVE approach. In 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009) (pp. 450-459). Springer Berlin Heidelberg.

[4] Palau, M., Ceccaroni, L., Gómez-Sebastià, I., Vázquez-Salceda, J., and Nieves, J. C. (2010). Coordination and Organisational Mechanisms Applied to the Development of a Dynamic, Context-aware Information Service. In ICAART (2) (pp. 88-95).

[5] Alvarez Napagao, S., Gómez-Sebastià, I., Vázquez-Salceda, J., and Koch, F. (2010). conciens: Organizational awareness in real-time strategy games. Proceedings of the 13th International Conference of the Catalan Assotiation of Artificial Intelligence. IOS Press, 2010, p. 69-78.

[6] Gómez-Sebastià, I., Garcia-Gasulla, D., Barrué, C., Vázquez-Salceda, J., and Cortés, U. (2010, August). A Flexible Agent-Oriented Solution to Model Organisational and Normative Requirements in Assistive Technologies. Proceedings of the 13th International Conference of the Catalan Assotiation of Artificial Intelligence. IOS Press, 2010, p. 79-88.

[1] `http://www.cs.upc.edu/~igomez/`

[7] Alvarez-Napagao, S., Koch, F., Gómez-Sebastià, I., and Vázquez-Salceda, J. (2011). Making games alive: an organisational approach. In Agents for games and simulations II (pp. 179-191). Springer Berlin Heidelberg.

[8] Palau, M., Gómez-Sebastià, I., Ceccaroni, L., Vázquez-Salceda, J., and Nieves, J. C. (2011). A framework for the development and maintenance of adaptive, dynamic, context-aware information services. In Agents and Artificial Intelligence (pp. 227-239). Springer Berlin Heidelberg.

[9] Gómez-Sebastià, I., Alvarez-Napagao, S., and Vázquez-Salceda, J. (2011). A Distributed Norm Compliance Model. Frontiers in Artificial Intelligence and Applications, Volume 232, pp. 110-119.

[10] Gómez-Sebastià, I., Garcia-Gasulla, D., Barrue, C., Vázquez-Salceda, J., and Cortés, U. (2012). Alive meets share-it: An agent-oriented solution to model organisational and normative requirements in assistive technologies. In Electronic Healthcare (pp. 319-326). Springer Berlin Heidelberg.

[11] Alvarez-Napagao, S., Gómez-Sebastià, I., Panagiotidi, S., Tejeda-Gómez, A., Oliva, L., and Vázquez-Salceda, J., (2012). Socially-aware emergent narrative. In Agents for Educational Games and Simulations (pp. 139-150). Springer Berlin Heidelberg.

[12] Gómez-Sebastià, I., Alvarez-Napagao, S., Vázquez-Salceda, J., and Felipe, L. O. (2012). Towards Runtime Support for Norm Change from a Monitoring Perspective. In Proceedings of AT2012 (pp. 71-85).

[13] Gómez-Sebastià, I., Garcia-Gasulla, D., Alvarez-Napagao, S., Vázquez-Salceda, J., and Cortés, U. (2012). Towards an implementation of a social electronic reminder for pills. In VII Workshop on Agents Applied in Health Care, A2HC 2012 (pp. 61 -71).

[14] Gómez-Sebastià, I., Alvarez-Napagao, S., and Vázquez-Salceda, J. (2013). Towards Heuristic Based Mobility Policy Optimisation. In Artificial Intelligence Research and Development: Proceedings of the 16th International Conference of the Catalan Association for Artificial Intelligence, Vol. 256 (pp. 297-300).

[15] Moreno, J., Cortés, U., Garcia-Gasulla, D., Gómez-Sebastià, I., and Alvarez-Napagao, S. (2013). Applying COAALAS to SPiDer. In Artificial Intelligence Research and Development: Proceedings of the 16th International Conference of the Catalan Association for Artificial Intelligence, Vol. 256 (pp. 326-335).

[16] Gómez-Sebastià, I., Alvarez Napagao, S., Garcia Gasulla, D., and Cortés U. (2013). Situated agents and humans in social interaction for elderly healthcare: the case of COAALAS. "Proceedings of the VIII workshop on Agents Applied in Health Care". Murcia: 2013, p. 105-119.

[17] Vázquez-Salceda, J., Alvarez Napagao, S., Tejeda J. A., Oliva Felipe, L. J., Garcia Gasulla, D., Gómez-Sebastià, I., and Codina Busquet, V. (2014). Making smart cities smarter using artificial intelligence techniques for smarter mobility. "SMARTGREENS 2014: proceedings of the 3rd International Conference on Smart Grids and Green IT Systems". SciTePress, 2014, p. IS7-IS11.

[18] Gómez-Sebastià, I. and Oliva-Felipe, L. and Alvarez-Napagao, S. and Garcia-Gasulla, D., Tejeda-Gómez, A., and Vázquez-Salceda, J., *A functional approach for disruptive event discovery and policy monitoring in mobility scenarios*, 7th European Lisp Symposium, 2014.

[19] Gómez-Sebastià, I. and Cortés, U. and Vázquez-Salceda, J., *Dynamic electronic institutions for patient treatment adherence monitoring*, In CCIA 2015 (pp. 267-270).

Other conference and workshop publications

[20] Chieppa, A., Gibert, K., Gómez-Sebastià, I., Sànchez-Marrè and M. (2008, July). Improving Pseudobagging techniques. In CCIA 2008 (pp. 161-169).

[21] Nieves, J. C., Gómez-Sebastià, I. (2009). Extension-Based Argumentation Semantics via Logic Programming Semantics with Negation as Failure. In LANMR (pp. 31-45).

[22] Osorio, M., Nieves, J. C., and Gómez-Sebastià, I. (2010). CF2-extensions as Answer-set Models. In COMMA 2010 (pp. 391-402).

[23] Gómez-Sebastià, I. and Nieves, J. C. (2010, August). WizArg: Visual Argumentation Framework Solving Wizard. Proceedings of the 13th International Conference of the Catalan Assotiation of Artificial Intelligence. IOS Press, 2010, p. 249-258.

[24] Garcia Gasulla, D., Alvarez Napagao, S., Tejeda, J. A., Oliva Felipe, L., Gómez-Sebastià, I., Béjar Alonso, J., and Vázquez Salceda, J. (2014). Social network data analysis for event detection. "ECAI 2014: 21st European Conference on Artificial Intelligence: 18-22 august 2014, Prague, Czech Republic: proceedings". Prague: IOS Press, 2014, p. 1009-1010.

[25] Béjar Alonso, J., Gómez-Sebastià, I., Alvarez Napagao, S., Tejeda, J. Garcia Gasulla, D.,Oliva Felipe, L., and Vázquez Salceda, J. (2014). Discovery of spatio-temporal patterns from location based social networks. Artificial Intelligence Research and Development: Recent Advances and Applications, 269, 126.

[26] Alvarez-Napagao, S., Tejeda-Gómez, A., Oliva-Felipe, L., Garcia-Gasulla, D., Codina, V., Gómez-Sebastià, I., and Vázquez Salceda, J. (2015). Urban context detection and context-aware recommendation via networks of humans as sensors. In Agent Technology for Intelligent Mobile Services and Smart Societies (pp. 68-85). Springer Berlin Heidelberg.